Titre: Title:	Routage par répartition de calcul et de trafic dans les réseaux de capteurs sans fil
Auteur: Author:	Éva-Maria Garcia
Date:	2006
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Garcia, É.-M. (2006). Routage par répartition de calcul et de trafic dans les réseaux de capteurs sans fil [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7885/

UNIVERSITÉ DE MONTRÉAL

ROUTAGE PAR RÉPARTITION DE CALCUL ET DE TRAFIC DANS LES RÉSEAUX DE CAPTEURS SANS FIL

ÉVA-MARIA GARCIA
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE
MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AOÛT 2006

Your file Votre référence
ISBN: 978-0-494-19302-0
Our file Notre référence
ISBN: 978-0-494-19302-0

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ROUTAGE PAR RÉPARTITION DE CALCUL ET DE TRAFIC DANS LES RÉSEAUX DE CAPTEURS SANS FIL

présenté par : Éva-Maria Garcia
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury constitué de :

M. BILODEAU Guillaume-Alexandre, Ph.D., président.
M. PIERRE Samuel, Ph.D., membre et directeur de recherche.
M. QUINTERO Alejandro, Doct., membre et co-directeur de recherche.
M. CHAMBERLAND Steven, Ph.D., membre.

À ces êtres chéris : mes parents et ma soeur, Simon, et Marisa.

Remerciements

Je voudrais remercier mes directeurs de recherche, M. Samuel Pierre et M. Alejandro Quintero, pour leur soutien et leur disponibilité lorsque j'en avais besoin. Aussi, je voudrais souligner le soutien de Marc Doyon pour régler les problèmes informatiques qui surviennent à tout moment.

Au plan plus personnel, je voudrais remercier Simon, toujours encourageant et foisonnant d'idées utiles ou rigolottes pour régler mon tout dernier problème de simulation. Je voudrais remercier mes supporters de longue haleine : mes parents, Pilar et Manuel, les deux êtres les plus travaillants, inspirants et généreux que je connaisse, ainsi que ma soeur, Isabel, un petit concentré d'enthousiasme et d'énergie.

Au LARIM, je voudrais remercier Moloud, pour sa rigueur inspirante et son amitié, Oscar pour sa compagnie dans les derniers milles ainsi que les trois mousquetaires Georges, Angelo et Sébastien-Maxime pour leur humour.

En tout dernier lieu, je voudrais remercier mes superviseurs chez Matrox qui m'ont donné le temps nécessaire pour boucler le tout à la fin ainsi que Charles Gagnon qui m'a généreusement cédé son ordinateur. À tout ce beau monde, merci !

Résumé

La localisation de cibles dans les réseaux de capteurs sans fil s'est traditionnellement effectuée par une approche centralisée. Les nœuds de détection transmettaient leurs données brutes à un serveur performant, qui se chargeait du travail de localisation. Avec la miniaturisation de la technologie, il est maintenant possible de doter les nœuds de détection de plus de capacités de calcul et de mémoire. Il devient alors possible de faire le traitement de localisation à même le nœud de détection. Cela peut même devenir souhaitable lorsque le nœud souhaite prendre action suite à la détection d'une cible, en déclenchant une alarme par exemple. Acheminer les données brutes jusqu'au serveur puis réacheminer les résultats de localisation au nœud détecteur implique un trafic important et superflu. Il serait utile que les nœuds détecteurs partagent entre eux leurs données de détection pour effectuer localement la localisation.

RCCT (Répartition de charge de calcul et de trafic), le protocole implémenté et simulé dans ce mémoire, propose un routage proactif basé sur OSPF-wireless. En utilisant des métriques pour refléter la qualité des liens et la charge de calcul de chacun des nœuds lors de la formation de la table de routage, les paquets de données de détection sont acheminés par les nœuds les plus sûrs aux nœuds les plus performants pour le travail de localisation. De plus, des grappes sont formées pour gérer le traitement des données, où des têtes de grappe se chargent du traitement de localisation et nomment des chargés de traitement lorsqu'elles ne suffisent plus à la demande de localisation de cibles.

Afin d'établir la validité de la solution RCCT, des simulations sur Qualnet sont réalisées. L'objectif des simulations est de comparer les performances du protocole RCCT à AODV. Les simulations sont réalisées dans un réseau de 25 nœuds détecteurs, pour un nombre de cibles variable et ont une durée de 20 minutes. Divers éléments sont observés notamment le nombre de localisations de cibles, le nombre moyen de paquets de contrôle et de paquets de données utiles ainsi que le délai moyen pour une localisation. On étudie les performances pour l'acheminement des données de localisations vers un serveur ainsi que vers un nœud détecteur. On étudie également l'utilité des chargés de traitement ainsi que des nœuds limitrophes pour le protocole RCCT.

Les résultats démontrent que lorsque les données de localisations sont destinées aux nœuds détecteurs, RCCT offre de meilleures performances qu'AODV. Lorsqu'elles sont destinées au serveur, AODV demeure plus performant. RCCT est donc un protocole approprié dans un contexte de réactivité, lorsqu'un nœud détecteur doit agir rapidement suite à la détection d'une cible.

Abstract

Wireless sensor networks have become a popular technology for target tracking. Determining the location of a target is done through triangulation. This method requires three detector nodes to share the target's detected information in order to determine its current position. Traditionally, all detector nodes would send raw target data to a powerful server that would determine the target's position and log it. With the miniaturization trend affecting technology, detector nodes have now more computing power and memory. As a result, it is now possible to compute the target's position directly at the detector node. In fact, it is useful if the node needs to act upon a positive localization of the target, by triggering an alarm for example. Sending all raw data to a server to then send back the results to the detector node generates redundant traffic on the network. Having a protocol that enables sharing raw target data between detector nodes for local treatment of the data would be useful.

RCCT (Répartition de charge de calcul et de trafic) , the protocol developed and simulated in this thesis, is based on OSPF-wireless. It uses metrics for the routing table calculation based on the quality of the links connecting nodes and based on the workload of the nodes. Hence, the raw target data travels through the most reliable nodes to the most efficient nodes for target localisation calculation. Also, clusters are used to manage the network. The cluster heads are in charge of localisation calculations and in charge of choosing the best nodes to help in this task when the heads become overloaded.

To validate RCCT, simulations were performed using the Qualnet network simulator. The simulations were to compare the performances of AODV and RCCT; they were carried out for a network of 25 nodes, for a varying number of targets and lasted 20 minutes each. The variables measured included the number of localised targets, the mean number of control and data packets and the mean time for a localisation. The difference between choosing the server as the final data destination, or choosing the detector node, is assessed in terms of protocol performance. Also, the usefulness of using nodes to help the cluster heads in data treatment is assessed.

The results show that RCCT is more efficient when the target localisation data is destined to the detector node. On the other hand, AODV is a more logical choice

if the data is destined to a server. Hence, RCCT is a useful protocol in a context of proactive networks, when the detector node uses the localisation data for concrete and quick action.

Table des matières

Liste des tableaux

Liste des figures

Liste des sigles et abréviations

AODV	Ad hoc On-demand Distance Vector
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
ECMP	Hash-Threshold Equal-Cost Multi-Path
ETX	Expected Transmission Count
GAF	Geographical Adaptive Fidelity
IGP	Interior Gateway Protocol
LBAR	Load Balanced Ad hoc Routing
LCA	Linked Cluster Algorithm
LEACH	Low Energy Adaptive Clustering Hierarchy
LQSR	Link Quality Source Routing
LSA	Link State Advertisement
LSF	Link State Flood
MANET	Mobile Ad hoc Network
MPR	Multi-Point Relay
OSPF	Open Shortest Path First
PEGASIS	Power-Efficient Gathering in Sensor Information Systems
RCCT	Répartition de charge de calcul et de trafic
RFC	Request For Comments
SPIN	Sensor Protocols for Information via Negotiation
TBRPF	Topology Broadcast based on Reverse-Path Forwarding
TDOA	Time Difference of Arrival
TND	TBRPF Neighbour Discovery

CHAPITRE 1

Introduction

Les réseaux de capteurs ont connu une importante évolution récemment. Plusieurs facteurs y ont contribué, notamment la miniaturisation des circuits électroniques de traitement et de capture de données ainsi que le développement des communications sans fil. Les réseaux sans fil ressemblent aux réseaux filaires, malgré certaines différences. Notamment, le medium sans fil ne permet pas les mêmes débits et fiabilité. Les réseaux de capteurs sans fil se distinguent par le fait qu'ils sont composés d'un réseau de nœuds dont l'objectif principal est de recueillir des données sur l'environnement à l'aide d'un ou plusieurs capteurs. Ces données sont alors transmises par la technologie sans fil dont sont pourvus les nœuds, que ce soit Bluetooth ou WLAN, vers un centre de traitement pour une intégration des données. Un grand effort de recherche s'est ainsi concentré sur les applications propres à ce champ. On y retrouve notamment des applications de détection et de suivi tant dans le domaine militaire que dans les situations d'urgence. Afin d'accélérer le processus de détection et de réaction des nœuds, il peut être intéressant de situer le traitement des données captées à même les nœuds formant le réseau plutôt qu'au centre de traitement. Le présent mémoire porte sur un protocole de routage pour les réseaux de capteurs utilisant le traitement réparti des données pour une réactivité plus rapide. Dans ce chapitre d'introduction, nous allons présenter les définitions et concepts de base nécessaires à l'exposé des éléments de la problématique ; par la suite, nous préciserons les objectifs de recherche et introduirons une esquisse méthodologique pour terminer avec le plan du mémoire.

1.1 Définitions et concepts de base

Dans les réseaux filaires, la répartition du traitement de tâches se fait souvent à l'aide du mode client-serveur. L'objectif est d'utiliser des machines dédiées dont les ressources sont optimisées pour traiter une tâche particulière. Ainsi, un usager transmet ses données brutes à ces machines et reçoit des données traitées. Aussi, on

peut utiliser des machines dédiées lorsqu'elles sont pourvues d'une capacité de traitement largement supérieure aux unités environnantes. Les réseaux de capteurs mobiles fonctionnent principalement à l'aide du modèle client-serveur. Tous les nœuds tentent alors d'acheminer l'information captée vers un centre de traitement. Cette technique est très friande d'énergie car l'information doit traverser tout le réseau pour parvenir au centre de traitement. Cette demande d'énergie est allégée par plusieurs protocoles, tel SPIN (Al-Karaki et Kamal, 2004). Certains protocoles prennent, quant à eux, une direction nouvelle en utilisant l'approche par agent mobile ou un agent visite les nœuds pour recueillir l'information (Xu et Qi, 2004). Cette approche consiste en un agent, comprenant du code exécutable, qui visite les nœuds, recueille l'information et la traite directement sur le nœud. Cette approche, justifiable pour des grands réseaux, est intéressante mais amène des questions de sécurité pour les nœuds et les agents mobiles.

Les protocoles ad hoc, incluant AODV, ont d'abord été retenus pour les applications de capteurs (Perkins *et al.*, 2003). Ils se déclinent en plusieurs familles : les protocoles actifs, réactifs, géographiques ou hiérarchiques. Les protocoles actifs, par leur connaissance de la topologie complète du réseau, se prêtent bien aux réseaux à faible mobilité. Les protocoles réactifs, formant des routes sur demande, se prêtent bien aux applications à plus haute mobilité et variabilité. Les protocoles géographiques sont intéressants pour les applications où la position géographique peut être exploitée. Les protocoles hiérarchiques, quant à eux, sont principalement réservés aux applications extensibles et au déploiement rapide.

Les réseaux de capteurs ont certaines caractéristiques qui diffèrent cependant des nœuds associés aux réseaux ad hoc : assistants personnels et ordinateurs portables. Les réseaux de capteurs ont présentement une mobilité généralement plus réduite car ils sont placés à un endroit stratégique afin de récolter des données. De plus, la mémoire et la capacité de traitement sont inférieures. Les capacités limitées des capteurs sont compensées par leur prix réduit, ce qui permet d'en acquérir une grande quantité et permet de la redondance.

Une des techniques envisageables pour améliorer l'efficacité des communications dans les réseaux sans fil de capteurs est l'utilisation de la répartition de charge de calcul. C'est d'ailleurs un sujet abordé par les thèmes d'ingénierie du trafic. Elle est souvent associée à la qualité de service. La qualité de service consiste à la différentiation des types de paquets et associe des traitements préférentiels à certains types de pa-

quets. L'ingénierie du trafic consiste à utiliser des techniques statistiques comme la théorie des files d'attente pour prévoir et modifier le trafic dans les réseaux de télécommunications. L'objectif de cette ingénierie est de répartir le trafic sur le réseau afin d'améliorer différents indices de performance, tels le délai moyen bout en bout, le délai maximal ou encore le taux de perte de paquets. En effet, la répartition de charge de calcul se définit par la répartition du traitement des données sur plusieurs nœuds, l'objectif visé étant l'obtention plus rapide d'information traitée. Les critères gérant cette répartition sont représentés par des métriques. Ces métriques qualifient la qualité d'un nœud voisin pour l'accomplissement d'une tâche ou encore qualifient la qualité d'un lien dans le but d'assister le nœud dans le choix des routes.

1.2 Éléments de la problématique

Plusieurs défis restreignent les capacités des réseaux de capteurs. Les aspects les plus abordés sont : la limite d'énergie des batteries ainsi que la faible et variable qualité des liens sans fil. Ces problèmes sont abordés dans de nombreux articles qui proposent des techniques afin de limiter leur impact (Al-Karaki et Kamal, 2004). Les principales techniques découvertes sont l'agrégation ou fusion de données pour diminuer le trafic, l'utilisation de grappes, ou encore l'utilisation du mode dormant pour économiser de l'énergie. Plusieurs algorithmes ont également été développés pour gérer les collisions de paquets et pour assurer le suivi de l'état des nœuds capteurs (Akyildiz *et al.*, 2005).

D'autres problèmes pourtant cruciaux ont été adressés dans une moindre mesure. On y retrouve la répartition de charge de calcul et la répartition de charge de trafic. Ces champs ont été peu abordés car les capacités des réseaux de capteurs sont généralement très limitées. On constate cependant que les nœuds pour certaines applications ne sont pas restreints dans la même mesure par ces critères et disposent de plus grandes réserves d'énergie. De plus, l'évolutivité des capteurs permettra bientôt une plus grande autonomie sur tous ces aspects. Certains protocoles, tel LBAR, se penche sur la problématique de la répartition de trafic pour les réseaux ad hoc. Cependant, ces protocoles répartissent le trafic dans un contexte de trafic général et non pas dans un contexte de trafic orienté vers le calcul réparti.

La répartition de charge de calcul permet que les données brutes soient ainsi envoyées à des nœuds voisins qui se chargent de traiter l'information. Les réseaux de

capteurs ont généralement des capacités de traitement moins évoluées que le centre de traitement auquel ils se rattachent. Cependant, l'envoi jusqu'au centre de traitement de toute l'information est une lourde tâche pour les nœuds et il est utile de traiter l'information près des nœuds s'ils doivent agir sur le résultat du traitement de l'information captée. Un exemple d'application est notamment la détection d'alarmes. Il devient donc nécessaire d'établir des stratégies de routage de l'information afin de permettre aux nœuds de faire le traitement requis de manière optimale. De plus, en assumant que tous les nœuds sont identiques, il est important de tenir compte des capacités des nœuds avoisinants à un moment donné afin de s'allier la collaboration des nœuds les plus avantageux. Ce fonctionnement viserait ainsi à s'éloigner de l'idée d'un centre de traitement, ou néanmoins à le reléguer à des tâches d'un autre ordre, comme par exemple pour du suivi ou pour une intégration des données à des applications externes.

Le défi est donc de maximiser le nombre de tâches accomplies par le réseau de capteurs en minimisant le nombre de collisions et le trafic généré.

1.3 Objectifs de recherche

L'objectif principal de ce mémoire est de concevoir un protocole de routage pour les réseaux sans fil de capteurs permettant de distribuer la charge de calcul sur les nœuds voisins en tenant compte de différentes métriques. De manière plus spécifique, ce mémoire vise à :

- Analyser les protocoles actuels de routage reliés aux réseaux de capteurs afin d'en extraire les caractéristiques principales mais aussi certaines lacunes ou possibilités d'amélioration dans le cadre de la conception d'un nouveau protocole de routage.
- Concevoir un protocole de routage répondant aux caractéristiques provenant de l'analyse mais qui, de plus, intègre des améliorations aux lacunes ou faiblesses observées dans les protocoles actuels. Notamment, le protocole de routage tentera de tirer parti des nœuds voisins selon leur taux d'utilisation afin de localiser et d'accélérer le traitement des données. Les nœuds voisins appelés à collaborer seront également choisis en fonction de métriques supplémentaires telles la qualité du lien sans fil.
- Implémenter le protocole à l'aide de l'outil de simulation Qualnet 3.9.

- Évaluer la performance du protocole dans une multitude de situations pour déterminer son efficacité.

1.4 Esquisse méthodologique et requis du protocole

L'étape initiale de ce travail consiste d'abord en bien définir le protocole à développer en s'appuyant sur les travaux menés dans le domaine. Le protocole devra permettre de répartir le traitement des paquets de données en vue possiblement d'agir sur les résultats obtenus. Les applications cibles devront être analysées afin d'ajuster les spécifications du protocole. Le protocole sera basé sur l'un des protocoles développés présentement et ayant des caractéristiques favorables pour notre application. Il sera ensuite modifié pour tenir compte de métriques spécifiques aux trois facteurs que nous avons isolés, soit la répartition de charge de calcul, de charge de trafic ainsi que la vérification de la qualité des liens. Ces métriques pourront s'appliquer avec des proportions différentes afin de mettre l'emphase davantage sur l'une ou l'autre. Le protocole pourra s'attarder notamment à l'impact d'éléments tels que :

- l'influence de la quantité de données à transmettre pour le traitement ;
- l'influence de l'ampleur de traitement à réaliser pour chaque tâche ;
- l'influence de la variété de tâches à traiter ;
- l'influence de la qualité des liens ;
- l'influence de la taille du réseau ;
- l'influence de la défaillance d'un lien.

Le protocole découvrira la topologie du réseau et devra maintenir à jour cette topologie malgré les liens défaillants, en optimisant la vitesse de récupération du réseau. L'utilisation de techniques d'agrégation d'information sera considérée tant pour le transport des données brutes comme pour les données traitées. L'objectif sera de démontrer que la répartition du traitement de tâches sur plusieurs nœuds permet une réactivité supérieure par rapport à l'envoi d'information vers un centre de traitement, tout en limitant le trafic maximal sur les nœuds ainsi que l'énergie requise pour ces transmissions.

1.5 Plan du mémoire

Ce mémoire se compose de cinq chapitres. Suivant ce chapitre introductif, le chapitre 2 propose une revue des protocoles développés dans le champ des réseaux ad hoc et spécifiquement des réseaux de capteurs ; il présente également les grands principes de la technologie 802.11, les techniques pour distribuer le calcul ainsi que pour distribuer le trafic sur divers chemins.

Le chapitre 3 présente le modèle développé pour la répartition de charge dans les réseaux de capteurs ; il explique en détail les différents éléments composant le protocole, tant au niveau de la gestion de topologie qu'au niveau des métriques développées.

Le chapitre 4 se penche sur l'implémentation du protocole et étudie les performances dudit protocole. Il le compare à des protocoles actuels et fait ressortir les points forts et les points faibles du protocole présenté. Il expose des performances quantitatives du protocole.

Le chapitre 5 conclut le mémoire en présentant une revue des résultats obtenus mis en contexte avec l'état de l'art, les limitations actuelles du protocole, ainsi que les voies à suivre pour pousser l'investigation vers des objectifs conjoints à ceux de ce mémoire.

CHAPITRE 2

Routage dans les réseaux Ad Hoc

Les réseaux de capteurs découlent directement des réseaux ad hoc, où les nœuds sont constitués généralement d'ordinateurs portables ou d'assistants personnels numériques. Une analyse des protocoles de routage pour les réseaux de capteurs débute donc par une étude des protocoles des réseaux ad hoc. Ce chapitre se penche notamment sur les solutions commerciales de routage offertes dans le domaine des réseaux ad hoc. Les lacunes de ces solutions ont été palliées par les protocoles développées par les laboratoires des recherche. Notamment, nous introduirons et analyserons des protocoles membres des grandes familles suivantes du routage ad hoc : routage passif et routage proactif. Les caractéristiques plus importantes ou uniques seront mises de l'avant pour chacun des protocoles principaux de ces familles. Par la suite, une analyse comparative des protocoles ad hoc permettra d'identifier les forces et faiblesses de chacun et mettre l'emphase sur le besoin de protocoles de routage pour les réseaux de capteurs. Ces nouveaux protocoles, plus spécialisés, utilisent différentes stratégies afin d'optimiser un aspect particulier du problème de routage, que ce soit par la répartition du trafic ou par la division en grappes. Finalement, des concepts connexes, tels que les caractéristiques et faiblesses du protocole 802.11 et les méthodes de répartition de charge de trafic et de calcul, seront également introduits.

2.1 Protocoles de routage

La plupart des points d'accès disponibles sur le marché n'offre pas une option de routage multi-bonds dans le mode ad hoc. Certains fabricants spécialisés, tels que Firetide Networks, offre des solutions intégrées de communication incluant le routage multi-bonds. Par exemple, Microsoft offre dans ses pilotes une implémentation du protocole ad hoc nommé LQSR, qui tient compte de la qualité du lien pour choisir la route optimale. Un vaste effort de recherche est déployé depuis quelques années pour développer des protocoles de routage pour les réseaux ad hoc. Les réseaux ad

hoc représentent un défi de taille car ils présentent plusieurs caractéristiques rendant délicat le processus de routage. Notamment, les réseaux ad hoc sont sujets à la variabilité de la qualité du lien, aux contraintes de puissance d'alimentation, à la capacité des liens, à la variabilité du milieu physique mais surtout à la grande mobilité des usagers. L'investigation dans ce domaine a mené à plusieurs RFCs, publiés par l'IETF, en voie de devenir des standards. AODV, DSR et OSPF font partie des RFCs et *internet drafts* concernant les protocoles de routage ad-hoc. Basés sur ces protocoles, de multiples autres protocoles sont développés, mettant l'emphase sur l'un ou l'autre des aspects des réseaux ad-hoc.

2.1.1 Protocoles de routage WLAN actifs

Royer et Toh (1999) ont exposé les principaux protocoles régissant les communications ad hoc. Il existe une panoplie de protocoles actifs. Ceux-ci se distinguent par le fait que chaque nœud composant le réseau a une vision complète du réseau, bien qu'il n'ait pas besoin de communiquer avec tous les nœuds le composant. La représentation globale du réseau est mise à jour à intervalles réguliers.

OSPFv2

OSPF (Open Shortest Path First), l'un des protocoles de routage les plus répandus dans les réseaux filaires, a été l'un des candidats initiaux pour les réseaux ad hoc multi-bonds. Cependant, certaines caractéristiques du protocole se prêtent toutefois moins bien au contexte des communications sans fil.

OSPF est décrit en détail par un RFC (J.Moy, 1998) et est un protocole de routage TCP/IP pour l'internet. Il fait partie de la famille IGP (Interior Gateway Protocol) car il régit le routage dans un système autonome, sous une entité administrative et protocolaire uniforme ; il ne sert pas de medium entre différentes interfaces réseaux correspondant à des systèmes autonomes distincts. Ce protocole est basé sur les états des liens, à l'aide de l'algorithme de Dijkstra, et non pas sur les vecteurs de distance associés à l'algorithme de Bellman-Ford.

Afin de connaître l'état des liens des voisins, le nœud enverra des LSA (link state advertisement) ou des messages d'annonce d'état des liens. Il y a plusieurs types de LSA. Les LSA sont envoyés périodiquement mais aussi automatiquement lorsqu'un changement dans l'état du routeur se produit. L'ensemble des LSA reçus par un

routeur forme sa base de données d'états de liens.

En effet, comme tout protocole basé sur les états des liens, chaque routeur maintient une base de données décrivant la topologie du système autonome. À l'aide de cette base de données, chaque routeur construit un arbre des plus courts chemins où il tient la position de racine. Par la suite, cet arbre permet de générer la table de routage servant à acheminer les données utiles.

La découverte de voisins se fait à l'aide du protocole HELLO. Chaque nœud tentera ensuite de former des adjacences (adjacency) avec le nouveau voisin. Ces adjacences sont une forme de relation privilégiée permettant de synchroniser les bases de données d'états de liens afin d'assurer une topologie identique dans l'aire du système autonome. Pour certaines interfaces, on doit plutôt élire un routeur désigné. Celui-ci devient l'unique adjacence de tous les routeurs de l'aire et donc en charge de la synchronisation des bases de données.

De plus, OSPF présente d'autres caractéristiques intéressantes. Notamment, il permet de router les paquets IP en se basant uniquement sur l'adresse IP de destination, que l'on retrouve dans l'entête IP du paquet et n'a donc pas besoin d'une encapsulation additionnelle à l'intérieur du système autonome. Aussi, OSPF est un protocole de routage dynamique ; lors d'une défaillance au niveau de l'interface d'un routeur, un nouveau chemin sans boucle est automatiquement recherché et créé après un période déterminée de convergence. OSPF est également très extensible car il permet l'union de plusieurs groupes de réseaux, appelés aires. L'ensemble des aires forme le système autonome. Ainsi, la topologie d'une aire est cachée du reste du système autonome ce qui permet une gestion répartie.

OSPFv2 wireless

Un certain délai est requis avant que les nœuds d'une aire aient partagé l'ensemble du contenu de leur base de données. Il a été démontré que ce délai de convergence peut être trop long dans les réseaux ad hoc avec le protocole OSPF. Pour cette raison, Ahrenholz *et al.* (2004) ont proposé une version sans fil du protocole où l'on ajoute un nouveau type d'interface : l'interface sans fil. Cette modification à l'algorithme élimine le concept d'adjacences et de synchronisation complète des bases de données. Elle propose plutôt un mécanisme d'inondation de LSA dans le réseau, inondation sans garanties de réception et sans obligation d'accusés de réception.

En effet, les réseaux ad hoc ont la capacité d'envoyer des messages de diffusion

générale (broadcast) mais tous les nœuds ne sont pas en mesure de les recevoir. Effectivement, ces réseaux ne sont pas de connectivité complète, c'est-à-dire que les routeurs d'une aire n'ont pas tous un lien direct avec les autres routeurs formant l'aire. Pour cette raison, la recherche d'un routeur désigné peut ne jamais converger. De plus, pour une interface point à multipoints, le nombre d'adjacences requises, lorsque le réseau grandit, peut rapidement générer trop de trafic de contrôle.

À l'aide de la nouvelle interface sans fil implémentée dans ce protocole, il est possible d'envoyer un nouveau type de paquet de contrôle, le LSF (link state flood). Cette nouvelle interface ajoute aussi un rôle à certains routeurs, celui de relais multipoints. Un routeur est choisi comme relais multipoints par un autre nœud voisin afin qu'il diffuse à tous ses autres voisins les messages que ce dernier lui fait parvenir. Ainsi, les mises à jour en mode sans fil se font à l'aide des LSF. Finalement, les paquets HELLO permettent à tout membre d'une aire de connaître ses voisins symétriques ou asymétriques jusqu'à une distance de deux bonds.

TBRPF

TBRPF (Topology Broadcast based on Reverse-Path Forwarding) est dérivé de OSPF pour s'adapter aux contraintes des réseaux sans fil. Il prône aussi une vision complète de la topologie du réseau pour déterminer de nouvelles routes rapidement si des liens échouent.

Découverte de Topologie et Routage

Le RFC 3684 (Ogier *et al.*, 2004) explique le protocole et Green et Obaidat (2003) ont démontré les performances du protocole. TBRPF peut utiliser le protocole TND (TBRPF Neighbour Discovery) pour la découverte de topologie et cela indépendamment de la technique de routage choisie. Le TND consiste en l'envoi de messages HELLO différentiels. Ces messages HELLO ne sont envoyés que pour indiquer des changements de topologie. Le paquet HELLO est donc plus court et peut être envoyé plus fréquemment pour s'ajuster à une topologie changeante. À partir de ces paquets, les nœuds maintiennent la liste de nœuds unidirectionnels et de nœuds bidirectionnels.

Comme avec OSPF, chaque nœud maintient un arbre des plus courts chemins et envoie une vision partielle de sa topologie à ses voisins. Ce sous-arbre est envoyé périodiquement alors que les paquets HELLO se chargent d'envoyer les changements apportés à cette topologie à une fréquence supérieure. Le sous-arbre envoyé comprend

les liens utilisant le nœud comme dernier saut avant la destination.

Résultats

TBRPF envoie les mises à jour en suivant le chemin inverse d'un arbre de couverture minimum et non pas par diffusion générale. TBRPF a un coût de l'ordre de V où V sont les nœuds du graphe alors qu'OSPF a un coût de mise à jour de l'ordre de V^2 car la mise à jour implique non seulement l'envoi de sa propre table de routage mais implique aussi la diffusion des tables de routage reçues des voisins. Les résultats des tests de performance démontrent qu'en situation de trafic maximum, et pour une topologie complète de 8 nœuds, TBRPF permet une réduction de 75% du trafic de contrôle et jusqu'à 90% en présence d'un trafic typique.

DSDV

DSDV (Destination-Sequenced Distance Vector) est également un protocole proactif. Il utilise une métrique se basant sur le nombre minimum de sauts, ou plus court chemin, entre la source et destination. Le calcul des chemins les plus courts se fait à l'aide de l'algorithme de Bellman-Ford. Chaque nœud possède une table de routage où tous les nœuds du réseau sont inscrits, ainsi que le nombre de sauts pour s'y rendre et un numéro de séquence indiquant la validité de la route. La transmission de ces tables de routage se fait à travers un envoi complet (full dump) de la table ou un envoi partiel ne contenant que les modifications additionnelles apportées à la table depuis le dernier envoi complet.

2.1.2 Protocoles de routage WLAN passifs

Les protocoles réactifs sont présentés comme une amélioration aux protocoles proactifs car ils évitent que chaque nœud doive tenir compte de tout le réseau, et de tous les mouvements et changements qui s'y produisent, parce qu'il n'a à communiquer qu'avec un sous-ensemble de tous les nœuds. L'inconvénient par contre est qu'un chemin n'apparaissant pas à la table de routage doit être découvert d'abord et cela retarde l'envoi des paquets vers la destination. AODV et DSR seront présentés d'abord car ils servent de référence pour plusieurs protocoles.

AODV

Héritier de DSDV, AODV (Ad Hoc On Demand Distance Vector) a été développé afin de réduire la quantité de trafic de contrôle sur le réseau (Perkins *et al.*, 2003). Il est utile pour des réseaux allant de la dizaine aux milliers de nœuds. Il bâtit les routes sur demande à l'aide d'un mécanisme de demande de création de route par diffusion générale et de réponse à ces demandes. AODV se démarque aussi par ses mécanismes de maintenance et réparation de routes et sa procédure de redémarrage pour maintenir le réseau stable.

Ainsi, lorsqu'un nœud désire envoyer un message vers une destination pour laquelle il n'a pas d'entrée à sa table de routage, il envoie une requête de routage (RREQ) à ses voisins, qui font de même à leur tour et ainsi de suite. Chaque nœud recevant le RREQ l'utilise pour valider sa route vers le nœud précédant, ainsi que vers la source. Au besoin, ces entrées sont créées dans la table de routage et constitue le chemin de retour vers la source (reverse path). Lorsque la destination est atteinte, une réponse à la requête (RREP) est envoyée sur un chemin unique vers la source. Ce mode de réponse implique qu'AODV n'est fonctionnel que pour des liens symétriques. Un numéro de séquence est utilisé afin de déterminer la validité d'une route. Les numéros de séquence sont très importants dans AODV car ils évitent de former des routes contenant des boucles. Lorsqu'un nœud reçoit une réponse RREP, il inscrira à sa table le voisin la lui ayant fait parvenir comme étant le prochain nœud vers la destination (forward path).

La maintenance des routes doit, quant à elle, s'effectuer lorsqu'un nœud se déplace hors zone. Dans ce cas, le voisin précédant, pour chaque route, propage un message de notification de défaillance de lien (link failure notification). Ce message indiquant une défaillance est le RERR, ce qui signifie erreur de routage. Il est envoyé à tous les nœuds en amont, incluant la source, afin qu'il effacent l'entrée correspondant à cette route. De plus, une réparation locale peut avoir lieu lorsqu'un lien se brise sur une route active. Ainsi, le nœud en amont peut décider de tenter une réparation locale en découvrant une route alternative. Si cette démarche échoue, il doit envoyer un message de défaillance. Le redémarrage d'un nœud sous AODV requière un comportement distinct d'un démarrage. En effet, lorsqu'un nœud redémarre, il est possible que d'autres nœuds le référencent encore alors que le nœud lui-même a perdu sa table de routage et même son numéro de séquence. Afin d'éviter des conflits potentiels, le nœud devra attendre un temps déterminé afin que toutes les routes de voisins uti-

lisant ce nœud expirent et qu'elles soient effacées. Le nœud peut durant ce temps remplir sa propre table de routage à l'aide des messages AODV reçus. Après ce temps déterminé, il peut à nouveau émettre des messages AODV car aucun nœud n'aura de routes encore dépendantes de lui.

AODV établit des routes sur demande de façon efficace et possède même des moyens de pallier à la défaillance de liens par la réparation locale. L'usage des messages HELLO par diffusion générale est optionelle et permet à un nœud de s'assurer de la présence et de l'opérativité d'un voisin et de découvrir de nouveaux voisins. En somme, AODV peut être beaucoup plus léger qu'un protocole proactif, surtout lorsqu'un nombre restraint de routes est utilisé.

DSR

DSR (Dynamic Source Routing) est aussi un protocole réactif. La découverte de route se fait, comme avec AODV, lorsque la cache de routage n'inclut pas de route vers la destination. La découverte de route se fait par diffusion générale d'une requête de route. Si le nœud n'a pas déjà reçu la requête, il la renvoie par diffusion générale à ses voisins après avoir ajouté son Id à la liste (route record) contenu dans le message. Une différence par rapport à AODV est que, lorsque le nœud parvient à la destination, la requête contient tous les nœuds parcourus pour s'y rendre. Cette liste (route record) est ajoutée à la réponse à la requête. La réponse utilisera cette liste pour être acheminée jusqu'à la source si les liens sont symétriques. S'ils ne le sont pas, une nouvelle découverte de route sera initiée et on ajoutera la route demandée originellement (piggyback) à la nouvelle requête de route vers la source. La maintenance de route se fait à l'aide de message d'erreurs de routage (route error packets) ainsi qu'avec des messages d'accusé de réception. Les premiers sont envoyés à tous les liens inclus dans une route affectée par le lien défaillant, les deuxièmes sont envoyés optionnellement pour s'assurer que les voisins sont en opération. On appelle accusé de réception passif lorsque le voisin écoute les transmissions de ses voisins.

LBAR

Le protocole LBAR (Load Balanced Ad hoc Routing), présenté par Zhou et Hassanein (2001), est un protocole réactif dont l'objectif est de répartir de manière optimale le trafic sur les nœuds disponibles. Il a été démontré que le délai de transmission d'un

paquet dans un réseau ad hoc augmente lorsque le réseau est de faible mobilité. Ce fait surprenant se doit au choix d'un nombre limité de routes optimales devant supporter tout le poids du trafic. En effet, sous LBAR, la découverte de route se distingue des protocoles vu jusqu'à maintenant par l'usage de métriques. Elle consiste en l'envoi d'un message de diffusion générale où le coût entre la source et le nœud récepteur est stipulé par une métrique. Ce processus est répété par les voisins et le message atteint finalement le nœud destination. Le nœud destination, après un certain temps déterminé, choisira le chemin ayant le meilleur coût. Ce coût est basé sur le degré d'activité de chaque nœud. Le nœud source sera informé de la route choisie par un accusé de réception acheminé sur la route en question. En cas de bris de liens, le nœud destination est informé et il achemine une route alternative au nœud qui a découvert le bris. Les messages HELLO sont, comme pour AODV, utilisés pour la gestion de connectivité.

La métrique déterminant le coût d'un lien est donc définie par le nombre de routes actives utilisant ce lien. Les routes actives sont les routes entre sources et destinations qui envoient activement des paquets. Contrairement aux réseaux filaires, les réseaux sans fil se caractérisent par des délais de transmission qui ne sont pas dus uniquement à un nœud congestionné mais aussi aux voisins de ce nœud se trouvant dans un état congestionné. Cette situation se définie comme de l'interférence de trafic. La fonction de coût se définit ainsi par :

$$C_k = \sum_{i \in k} \left(A_i + \sum_{i \in k} A_j^i \right) \tag{2.1}$$

où A_i sont les routes actives sur le nœud i et $\sum_{i \in k} A_j^i$ est l'interférence de trafic des routes actives sur les nœuds voisins. Le protocole LBAR se démarque pour certains types de réseaux. Notamment, pour un réseau de 50 nœuds sous un trafic nominal de 2 Mbps, et avec de 10 à 40 sources de trafic à 4 paquets/sec., le protocole LBAR produit des délais plus petits qu'AODV et DSR. Notamment, les délais augmentent avec 40 sources de trafic avec une mobilité faible pour AODV et DSR mais ce délai est constant sous LBAR. Lorsque la mobilité est forte, LBAR présente des résultats similaires à AODV et DSR car le trafic se répartit plus également sur les liens actifs.

2.1.3 Analyse - protocoles proactifs et réactifs

Plusieurs différences existent entre les protocoles réactifs et proactifs. La différence fondamentale est bien sûr que les protocoles proactifs possèdent une vision complète de la topologie du réseau et que les protocoles réactifs établissent les routes selon leur besoin. Notamment, DSDV envoie périodiquement des mises à jour ce qui le rend lourd car ces mises à jour sont envoyées en présence ou non de changements. AODV et DSR se ressemblent quant à la découverte de routes, cependant le surdébit est plus important avec DSR car chaque paquet y inclut la route complète plutôt que seulement l'adresse de destination. De plus, la demande en espace de mémoire est plus élevée avec DSR car chaque nœud garde la route complète alors qu'AODV ne garde que le prochain nœud. AODV contient aussi des capacités pour l'envoi par multi-diffusion. Le désavantage d'AODV est qu'il requière des liens symétriques. DSR est pensé pour des réseaux de taille réduite à faible mobilité. Il a l'avantage de prévoir de multiples routes vers une destination, ce qui est utile en cas de défaillance d'un lien, et ne requière pas l'envoi périodique obligatoire de mises à jour. Les deux protocoles AODV et DSR permettent la réparation locale de routes. Les protocoles proactifs et réactifs ont évolué et pris diverses formes pour remplir les exigences des réseaux de capteurs. Les protocoles suivants se subdivisent à nouveau en différentes catégories : protocoles de négociation, basés sur les requêtes, basés sur la cohérence, basé sur les multiples chemins.

2.1.4 Protocoles de routage maillés et grappes

Akyildiz *et al.* (2005) se sont penchés sur les récents progrès de la recherche sur les réseaux de capteurs sans fil. Leur analyse a mis à jour qu'une des limitations actuelles de ce type de réseau est le manque d'extensibilité. En effet, les performances diminuent drastiquement lorsque le nombre de nœuds ou de bonds augmente. L'article recommande donc d'ajuster les protocoles MAC et de routage pour les réseaux maillés. Il spécifie que pour le protocole MAC et ses dérivés, il est difficile d'obtenir des résultats raisonnables lorsque le nombre de bonds est de 4 ou plus. Entre autres, l'usage du modèle CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) introduit une faible efficacité de réutilisation spatiale (spatial-reuse) ce qui limite l'extensibilité. Pour un trajet multi-bonds, les collisions à chaque nœud ont un effet d'accumulation qui se traduit par un débit faible de bout en bout. De

plus, lorsque le mécanisme de *carrier sense* virtuel est appliqué, cela implique un délai supplémentaire. Pour un réseau statique multi-bonds, il démontre que la puissance de transmission idéale pour les nœuds, pour maximiser la capacité de débit, est atteinte lorsque le nœud détecte six nœuds voisins. Afin de réduire la puissance de communication, deux stratégies sont présentées : la première consiste en l'ajout de nœuds exclusivement de relais ; la seconde suggère de regrouper les nœuds en grappes et d'utiliser des connections filaires pour passer d'une grappe à l'autre. On y recommande aussi le routage optimal par séparation de paquets ; ce qui implique le besoin de chemins disjoints. Cette approche est complexe mais permet une meilleure répartition de la charge.

Par rapport au routage par grappes, on indique qu'il faut définir une tête de grappe. Tout trafic inter-grappe transite par la tête de grappe. La tête de grappe devient une passerelle. On recommande l'utilisation de protocoles de routage réactifs dans la grappe et proactif au niveau inter-grappe. On souligne toutefois que la tête de grappe peut devenir un goulot d'étranglement. Un exemple de méthode d'élection pour les têtes de grappes, favorisant la répartition de charge et développé par Amis et Prakash (2000), est présenté ci-dessous.

Grappes pour la répartition de charge

Les têtes de grappes forment un squelette virtuel et sont utilisées pour router les paquets des nœuds appartenant à leur grappe. Elles sont élues selon des critères de connectivité. Ces têtes de grappes sont cruciales au fonctionnement du réseau et dépensent leur réserve d'énergie plus rapidement. Il est donc utile de disposer d'un algorithme d'élection permettant à tous les nœuds de devenir tête de grappe afin d'assurer une meilleure répartition de charge. L' algorithme sert spécifiquement pour des liens bidirectionnels. Au départ, les algorithmes pour élire les têtes de grappe, par exemple l'heuristique LCA (Linked Cluster Algorithm), n'utilisaient que des grappes à un saut des voisins appartenant à sa grappe. Par la suite, l'algorithme Max-Min permit d'étendre cette notion vers des grappes où les nœuds sont à d sauts de la tête. Cet algorithme est plus rapide mais génère beaucoup de trafic chez les voisins immédiats de la tête. Dans les deux cas, l'élection se basait sur le numéro d'identification des nœuds. D'autres algorithmes se basent sur le degré de connectivité. Ils tendent cependant à des élections à répétition si la topologie est très changeante. La méthode présentée souhaite combiner la stabilité de la méthode Max-Min à l'objec-

tivité de la méthode de degré de connectivité. On souhaite donc maximiser la durée d'une tête de grappe et donc la stabilité du réseau. Pour améliorer les méthodes basées sur les numéros d'identification, on définit un numéro d'identification virtuel pour chaque nœud. Le nœud avec le plus grand numéro virtuel est choisi comme tête de grappe. Les nœuds virtuels sont ajustés pour représenter l'éligibilité du nœud. À chaque cycle de l'algorithme, ils sont modifiés. Après un temps déterminé comme tête de grappe, le nœud devient un nœud normal et une autre tête de grappe est choisie. Pour améliorer les méthodes basées sur le degré de connectivité, on utilise une technique de répartition de charge basée sur la variation de degré nodal. Si cette variation passe au-dessus d'un seuil déterminé, on élit aussi une nouvelle tête de grappe. Cette méthode est testée avec au plus 600 nœuds. Différents paramètres sont modifiés ; notamment, la valeur du numéro virtuel lorsqu'une tête de grappe est déchue, ainsi que la méthode de calcul du temps alloué pour cette tête de grappe.

Les résultats de la recherche indiquent que, par rapport à la méthode Max-Min avec ou sans répartition de charge, les têtes de grappes ont duré en moyenne entre 14% et 28% plus longtemps. La variance est plus forte de 500% lorsque la répartition de charge n'est pas utilisée. Pour la méthode de degré nodal, avec ou sans répartition de charge, on obtient une amélioration de 200% de la durée des liens. La variance est cependant plus faible sans répartition de charge. Ceci est dû au fait que la méthode de degré nodal sans répartition de charge élit ses têtes de grappe pour une durée équivalent à une période d'échantillonage. Donc, même si elles sont toutes de même durée, avec une variance faible, ces têtes de grappe ont une durée trop courte pour être utiles. La méthode de degré nodal avec répartition de charge demeure donc une meilleure alternative.

Routage par grappes - LEACH, PEGASIS

L'algorithme LEACH (Low Energy Adaptive Clustering Hierarchy) présenté par Al-Karaki et Kamal (2004) est l'un des algorithmes de routage les plus populaires pour le routage par grappes. Cet algorithme permet l'extensibilité des réseaux ainsi que des communications pratiques. Il permet aussi de réduire le trafic vers la station de base, ou station de traitement, ainsi que de réaliser la fusion de l'information. LEACH choisit au hasard quelques nœuds comme têtes de grappe et redistribue ce rôle afin de maintenir un niveau énergétique similaire à travers les nœuds du réseau. De plus, l'information est comprimée par les têtes de grappes. LEACH utilise une

protocole liaison MAC TDMA/CDMA (Time Division Multiple Access/ Code Division Multiple Access) afin de réduire les collisions intra-cellulaires et inter-cellulaires. L'agrégation est performée périodiquement donc LEACH se prête mieux à un contexte de surveillance constante.

Le fonctionnement de LEACH se divise en deux phases qui se répètent constamment : l'initialisation et l'état stable. L'initialisation permet l'élection des têtes de grappe et l'état stable permet la transmission d'information. Afin de choisir les têtes de grappe, chaque nœud choisit un nombre, basé sur la ronde actuelle, sur son désir de devenir tête de grappe et sur le nombre de nœuds n'ayant pas été tête de grappes. Après leur élection, les nœuds diffusent leur nouveau rôle. Tous les autres nœuds décident alors à quelle tête s'associer à partir de la puissance de signal du message reçu. Une fois informée des nœuds lui étant assignés, la tête de grappe développe un horaire TDMA où une fente horaire est attribuée à chaque nœud associé. Durant l'état stable, chaque nœud transmet les paquets d'information à la tête de grappe. Finalement, après une période déterminée, on revient à l'étape d'initialisation.

LEACH présente certaines lacunes. Le protocole assume que tous les nœuds sont capables de communiquer avec la station de base au besoin et ont une puissance de calcul suffisante pour supporter différents protocoles MAC. De plus, aucun mécanisme ne garantit la distribution uniforme des têtes de grappes sur le réseau. L'élection dynamique de têtes de grappes ajoute aussi des charges indirectes à l'opération du réseau et consomme de l'énergie.

Afin d'améliorer LEACH, le protocole PEGASIS (Power-Efficient Gathering in Sensor Information Systems) met l'emphase sur la communication avec ses voisins les plus proches et ces voisins se relaient pour l'envoi de l'information. Chaque ronde est formée d'une chaîne de voisins se relayant pour l'envoi de l'information à la station de base. L'objectif de PEGASIS est double. Il consiste à réduire la consommation d'énergie par la collaboration des nœuds voisins ainsi que permettre la coordination locale entre nœuds voisins pour réduire le trafic. PEGASIS ne forme pas de grappes mais plutôt une chaîne de voisins se relayant jusqu'à la station de base. Les nœuds ajustent la puissance de leur signal pour ne communiquer qu'avec un seul nœud. La construction de la chaîne se fait à l'aide d'une méthode Greedy. Les résultats démontrent que cette méthode est deux fois plus performante que LEACH par la réduction des charges indirectes bien que des ressources sont requises pour ajuster dynamiquement la topologie afin de refléter les baisses d'énergie des nœuds.

2.1.5 Protocoles de routage basés sur la négociation - SPIN

Al-Karaki et Kamal (2004) ont présenté les protocoles les plus récents concernant les réseaux de capteurs sans fil. Ces protocoles recherchent tous, principalement, l'économie d'énergie afin de préserver la vie du réseau. Cette section présente les protocoles basés sur la négociation.

Le protocole SPIN (Sensor Protocols for Information via Negotiation) dissémine l'information recueillie par les capteurs vers tous les nœuds, sous l'hypothèse qu'il puisse s'agir d'un centre de traitement. Ce protocole assume que les nœuds sont présents en grande densité et qu'il y a donc une redondance dans l'information captée. Un nœud cherchera donc à ne propager que l'information non présente chez les nœuds voisins. La famille de protocoles SPIN utilise des algorithmes pour la négociation d'information et l'adaptation aux ressources.

La négociation est l'élément clé de SPIN. Les nœuds utilisant SPIN assignent un nom de haut niveau décrivant l'information captée (métadonnée) et réalise une négociation de métadonnées avant de procéder à la transmission de données brutes. La sémantique de la métadonnée est spécifiée par l'application. Le protocole SPIN se divise donc en trois étapes. D'abord, un nœud ayant de nouvelles données l'annonce à l'aide d'un message ADV (advertisement). Un nœud intéressé répondra par un message REQ (request) afin d'obtenir l'information, envoyée dans des messages DATA. Le voisin, après avoir acquis l'information, l'annoncera lui aussi à ses voisins.

La gestion d'énergie forme également parti de SPIN. Lorsque le nœud a une baisse dans ses réserves d'énergie, SPIN détecte ce niveau et adapte le fonctionnement du protocole. En effet, un nœud n'annoncera l'information captée que s'il se sent en mesure de compléter toutes les étapes requises par le protocole.

De plus, il existe plusieurs variantes de SPIN. Notamment, SPIN-EC inclut une heuristique additionnelle pour la conservation d'énergie. SPIN-RL, de son côté, est utile dans les environnements bruyants car le protocole s'ajuste en présence de liens instables.

2.1.6 Protocoles de routage basés sur la requête d'information - Diffusion Dirigée

La Diffusion Dirigée est présenté également comme un protocole visant la conservation d'énergie (Al-Karaki et Kamal, 2004). Ce protocole a été popularisé par son

paradigme d'agrégation de données. Toutes les données générées par les capteurs sont regroupées en paires 'attribut-valeur'. L'objectif de ce protocole est de combiner en chemin l'information provenant de différentes sources en éliminant les redondances, en minimisant le nombre de transmissions, et ce, afin de conserver l'énergie du réseau.

La diffusion dirigée se base sur les concepts de gradients d'information et des intérêts. Lorsqu'un nœud capture un évènement, il crée un gradient d'information dans son voisinage. Un gradient décrit la valeur d'un attribut ainsi qu'une direction. La station de base envoie ses requêtes d'information en transmettant des intérêts par diffusion générale. Un intérêt décrit une tâche à performer dans le réseau. Lorsqu'un intérêt correspond au gradient annoncé, des chemins sont déterminés en examinant les multiples chemins possibles. Seuls les meilleurs chemins sont encouragés afin de réduire la propagation du message, et ce, selon un police locale. Tous les nœuds sur le chemin vers la station de base sont en mesure de performer l'agrégation de données. Le protocole réduit la consommation d'énergie en utilisant des chemins optimaux, déterminés empiriquement, ainsi qu'en cachant et traitant l'information dans le réseau. Des modèles, ER(Event Model) model et RS(Random Sources) model, ont été développés pour simuler des topologies spécifiques du réseau.

2.1.7 Protocoles de routage WLAN géographiques - GAF

Al-Karaki et Kamal (2004) ont présenté plusieurs algorithmes de routage basés sur la localisation, dont notamment le protocole GAF (Geographical Adaptive Fidelity). Dans ce protocole, l'aire du réseau est d'abord divisée en zones fixes formant une grille virtuelle. Les nœuds d'une même zone collaborent et jouent différents rôles. Par exemple, un nœud peut être choisi pour demeurer éveillé alors que les autres nœuds de la zone dorment afin de conserver leur énergie. Ce nœud doit alors surveiller et communiquer avec la station de base au nom du groupe. Chaque nœud utilise sa position GPS pour s'associer à un point de la grille virtuelle. Lorsqu'un évènement se produit, le nœud choisi réveille les autres nœuds pour que ceux-ci capturent des données. Ces données sont ensuite acheminées à la station de base à travers le nœud choisi.

2.1.8 Analyse - nouveaux protocoles pour les réseaux de capteurs

Les protocoles récemment proposés offrent des approches variées. Alors que SPIN et la diffusion dirigée ont une structure plate, LEACH et PEGASIS préfèrent une architecture hiérarchisée. Tous ces protocoles utilisent l'agrégation de données mais seuls SPIN et la diffusion dirigée utilisent une forme de négotiation et sont basés sur la requête. LEACH introduit une complexité d'état supplémentaire lors de l'élection des têtes de grappe. Par contre LEACH et PEGASIS s'adapte mieux à des réseaux grandissants. Ces deux protocoles ne propose cependant qu'un seul chemin vers le centre de traitement alors que SPIN et la diffusion dirigée offre de multiples chemins, et, avec la diffusion dirigée, le chemin proposé est optimal.

2.2 Algorithmes du plus court chemin

Les protocoles proactifs doivent souvent, à partir des messages de contrôle échangés, déterminer la route optimale pour l'envoi de l'information utile. Souvent, cela passe par la constitution d'un arbre des plus courts chemins qui permettent d'établir un chemin vers les différents points du réseau.

Les deux algorithmes les plus utilisés sont ceux utilisant un vecteur de distance puis ceux utilisant l'état des liens, qui sont présentés dans (Inetdaemon, 2004). L'algorithme de vecteur de distance calcule le nombre de bonds entre source et destination. Il choisira le chemin ayant le moins de bons comme route optimale et RIP est un exemple de protocole utilisant cet algorithme. L'algorithme de Bellman-Ford est un exemple d'algorithme de vecteur.

Les algorithmes d'état de liens font le suivi de l'état et du type de liens (débit) et produisent une métrique calculée à partir de ces paramètres et de paramètres spécifiés par l'administrateur de réseau. La route optimale sera celle dont le coût, déterminé à partir de la métrique, est moindre. Un chemin dont les interfaces sont plus rapides mais où le nombre de bonds est supérieur peut être choisi comme étant le chemin optimal. La charge de calcul requise par ce type d'algorithme est plus élevée que pour les algorithmes de vecteurs de distance. L'algorithme de Dijkstra est un exemple d'algorithme d'états de liens et le protocole OSPF est un exemple de protocole l'utilisant.

En général, si plusieurs types d'interfaces sont présents, il vaut mieux choisir un algorithme d'états des liens, sinon l'algorithme de vecteur de distance suffira.

2.3 Chemins de coût équivalent et répartition de charge de trafic

Lorsque le protocole de routage offre plusieurs chemins de mêmes coûts pour une route, il faut faire un choix. Les algorithmes se penchent sur cette question car le choix répété d'une route peut engendrer un trafic important sur certains liens, en négligeant des liens moins utilisés. Deux articles nous proposent des solutions différentes. Hopes (2000) a proposé une méthode de hachage pour le choix de la route. Li *et al.* (2005) ont proposé un algorithme Round-Robin qui réduit l'intervalle de détection de trafic tout en optimisant la répartition de charge de trafic.

La méthode ECMP (Hash-Threshold Equal-Cost Multi-Path) avec limite de hachage est utilisée pour choisir la route. Le routeur choisit d'abord une clé en performant le hachage sur les champs d'entête du paquet à envoyer. Tous les prochains bonds possibles se voient assigner une région unique dans l'espace de clés. Le routeur utilise la clé calculée pour déterminer quelle région utiliser et enfin quel prochain bond choisir. Les deux aspects étudiés sont la performance et l'impact du changement de chemin pour un flot. L'algorithme utilise des régions d'égale grandeur pour chaque route. Si le résultat de la fonction de hachage est réparti uniformément, les flots devraient aussi être uniformes. En utilisant un tableau en mémoire pour représenter chacune des zones, l'algorithme peut être performant puisqu'il se résume à une division et une lecture en mémoire. Les changements de route durant un flot sont à éviter avec TCP afin de remettre les paquets du flot en ordre par la suite. Cependant, ça ne devrait pas se produire car la clé ne devrait pas varier pour un même flot, car les entêtes de tous les paquets formant le flot seront identiques, et donc les paquets sont assignés à une même route pour un même flot.

D'un autre côté, la méthode Round-Robin avec poids dynamiques est une variante du Round-Robin traditionnel qui consiste à assigner consécutivement une route différente à chaque paquet, à partir d'une liste de chemins de coûts équivalents. Cette méthode détermine l'intervalle optimal pour effectuer une détection du trafic sur les liens et utilise le concept statistique de variance afin de déterminer le niveau de

répartition de charge. La méthode surveille le trafic sur les liens afin d'assigner les nouveaux flots au lien adéquat. La variance v utilise la variable *load*, pour représenter le trafic sur chaque lien, et est mesurée de la façon suivante :

$$v_t = \frac{\sum_{i-1}^{n} \left(load_{t,i} - \overline{load_t}\right)^2}{n} \tag{2.2}$$

où

$$\overline{load_t} = \frac{\sum_{i=1}^{n} load_{t,i}}{n} \tag{2.3}$$

Des poids sont assignés à chacun des liens pour indiquer leur niveau de charge. Ils sont définis ainsi :

$$w_{t,i} = \frac{1}{[(load_{t,i}/k) + 1]} \tag{2.4}$$

où

$$k = \sqrt{\frac{\int_t v_t dt}{T}} \tag{2.5}$$

T étant la période totale de simulation et k est un coefficient représentant la variance de charge totale du système pour une période de temps. Selon la valeur de ce poids, comparé aux poids assignés aux autres liens, un lien recevra une partie proportionnelle de tâches, et ce jusqu'à la prochaine détection. En utilisant le théorème de Little et la théorie des queues pour un taux d'arrivée de distribution exponentielle, on obtient que l'intervalle idéal de détection de la variance se situe dans les limites suivantes :

$$\frac{\overline{a}k}{\overline{L} - \overline{a}} \leq t < \frac{\overline{a}n^2 k}{(n-1)\,\overline{L}} \tag{2.6}$$

où $\overline{L}$ est la moyenne des longueurs des tâches en unités de temps, $\overline{a}$ est la moyenne des intervalles d'arrivée des tâches, n est le nombre de liens et t est l'intervalle de détection. À l'aide de la variance mesurée entre les différents liens à l'aide de k, un intervalle de détection est déterminé. Cet intervalle de détection performe légèrement moins bien que les algorithmes 'Least Load' et 'Least Connections', deux méthodes populaires, mais il ne nécessite que 10% des détections requises par ces algorithmes. La méthode Round Robin de base, quant à elle, produit des résultats désavantageux surtout dans des situations de grand trafic.

2.4 Technologie 802.11

Afin de définir un protocole de routage pour les réseaux de capteurs sans fil, il faut bien connaître les caractéristiques du médium utilisé pour la communication : le medium sans fil. Dans ce cas-ci, les couches physique et liaison du médium sans fil pour la plupart des applications commerciales sont définies par le protocole 802.11. Brenner (1996) et IEEE-SA Standards Board (2003) décrivent les deux couches définies par ce protocole. En effet, la technologie sans fil sous 802.11 a des contraintes nouvelles, notamment :

- le medium ne possède pas de limites physiques observables au-delà desquelles les paquets sont garantis de ne pas être reçus ;
- le medium n'est pas protégé contre les interférences causées par des signaux extérieurs ;
- la communication n'est pas aussi fiable comme celle des réseaux filaires ;
- les topologies sont dynamiques ;
- la connectivité n'est pas nécessairement complète ;
- les caractéristiques de propagation varient avec le temps et peuvent être asymétriques.

Présentement, le standard définit une seule couche MAC qui peut échanger avec trois couches PHY potentielles :

- Frequency Hopping Spread Spectrum ;
- Direct Sequence Spread Spectrum ;
- Infrarouge.

En plus des tâches typiques des couches MAC, la couche MAC sous 802.11 gère aussi des fonctions telles que la fragmentation, les retransmissions de paquets et les accusés de réception.

La couche MAC se différencie dans le mode sans fil de l'Ethernet. Il définit deux méthodes d'accès : la fonction de coordination distribuée (DCF) et la fonction de coordination de points. Le DCF consiste en un CSMA avec évitement de collisions alors qu'Ethernet utilise CSMA avec détection de collisions. Ceci a dû être modifié pour le medium sans fil car les liens sont bidirectionnels avec alternance (half duplex) et la détection simultanée est impossible. Le fonctionnement du CSMA/CA est le suivant : la station désirant transmettre écoute le médium et, si celui-ci est occupé, la station différera la transmission à plus tard, sinon elle transmet. Il reste tout de

même un risque que deux stations croyant le médium libre transmettent à la fois et créent une collision. Pour réagir à ces situations, le récepteur envoie un accusé de réception après avoir exécuté un checksum du message reçu. En son absence, la couche MAC se charge de retransmettre après une période d'attente exponentielle aléatoire (exponential random backoff). Avant même de transmettre une première fois, le medium doit être libre pour une durée minimale de temps. Afin de diminuer la probabilité de collisions entre deux stations, le *carrier sense* virtuel peut être adopté. Un paquet envoie un paquet de contrôle RTS afin d'obtenir le contrôle du médium. S'il obtient une réponse affirmative CTS pour la durée demandée, il peut occuper le canal pour la durée en question.

Une autre adaptation du 802.11 pour le médium sans fil est la fragmentation. En effet, les paquets utilisés pour Ethernet sont typiquement d'une longueur de plus d'un millier d'octects. Cependant, il est préférable d'utiliser des paquets plus courts dans le médium sans fil, notamment car le taux de bits d'erreur est plus élevé. De plus, en cas de collision ou d'environnement bruyant, la retransmission d'un plus court paquet cause moins de surdébit. Par contre, si les paquets sont effectivement de plus d'un millier d'octets, le protocole implémente un mécanisme simple de fragmentation/réassemblage à la couche MAC.

La trame 802.11 ajoute l'encapsultation présenté ci-dessous :

Préambule	Entête PLCP	Information MAC	CRC

Dans cette trame, le préambule est associé à la couche PHY. L'entête PLCP contient l'information logique permettant à la couche PHY de décoder la trame. L'entête MAC se compose de 9 champs dont notamment les adresses du récepteur et transmetteur, un indicateur représentant le type de paquet et l'information utile. Le dernier champ est le checksum.

2.4.1 Limitations et mesures du Protocole 802.11

Ng et Liew (2004) ont étudié les limitations du protocole 802.11, notamment celles du CSMA/CA. On y démontre qu'en contrôlant le trafic sur les liens, il est possible de diminuer le taux de paquets perdus, l'instabilité créée par le re-routage et les problèmes d'inégalité (unfairness). Les deux limitations étudiées sont les nœuds cachés et les limitations de réutilisation spatiale.

Le haut taux de paquets perdus et le problème d'inégalité peuvent se produire

avec le 802.11. La Figure 2.1 l'illustre. En effet, le nœud 0 perçoit les nœuds 1 et 2. Par contre, le nœud 2 perçoit non seulement les nœuds 0 et 1 mais aussi les nœuds 3 et 4. Le nœud zéro, ne s'abstenant de communiquer que lorsqu'il entend les nœuds 1 et 2 transmettre, utilisera toute la capacité de bande passante non utilisée par les nœuds 1 et 2. Cependant, le nœud 2 qui doit aussi partager sa capacité avec les nœuds 3 et 4 se trouvera rapidement débordé et devra rejeter certains paquets.

En plus de l'inégalité, l'instabilité due aux re-routages peut être problématique. Lorsque le nœud 0 dépasse le nombre maximum de retransmissions d'un paquet vers le nœud 2, débordé, il estime que le lien vers le nœud 2 est brisé. Ce qui n'est pas le cas. Le nœud 2 est simplement occupé par une autre transmission. Le nœud 0 tente alors une nouvelle découverte de route vers sa destination. S'il n'existe pas de route alternative, le nœud 2 est éventuellement redécouvert et ce cycle recommence. Ce comportement génère de larges variations de débit et ce débit, consistant de nombreuses retransmissions, est redondant.

Le problème du nœud caché doit également être gardé à l'oeil. Si le nœud 0 tente de communiquer avec le nœud 2, il écoute le médium pour savoir s'il est utilisé. Il n'entendra cependant pas le nœud 4 transmettre au nœud 2, car celui-ci se trouve hors de portée du nœud 0, et le nœud 0 commencera à transmettre. Cela crée des collisions. Ce problème est partiellement réglé avec l'utilisation du *carrier sense* virtuel. Cependant, cette méthode échoue lorsque les nœuds cachés envoient à deux récepteurs différents. En effet, si le nœud 4 envoie au nœud 1, malgré l'utilisation du *carrier sense* virtuel, le nœud 0 enverra quand même au nœud 2. Le nœud 2, sentant le médium occupé par le début de transmission entre les nœuds 4 et 1, ne recevra pas correctement la transmission du nœud 0.

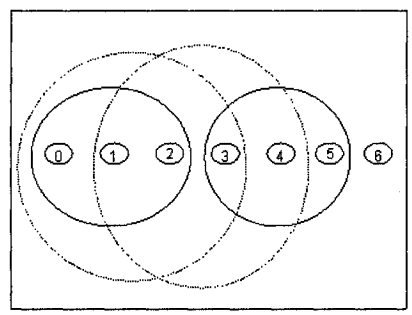

FIGURE 2.1 Zones de détection

Les auteurs déterminent une fonction pour trouver le débit optimal en tenant compte des nœuds cachés pour un problème où les nœuds sont disposés linéairement. Le débit optimal est déterminé par les formules suivantes :

$$T = x * (i - \rho)) * d * data\_rate \tag{2.7}$$

Où

$$\rho = \left(\frac{x}{1-2x}\right) a \tag{2.8}$$

$$x = \frac{(2+a) - \sqrt{a^2+2a}}{4+2a} \tag{2.9}$$

$$a = (PACKET) / (DIFS + PACKET + SIFS + ACK) \tag{2.10}$$

$$d = DATA / (DIFS + PACKET + SIFS + ACK) \tag{2.11}$$

et *data_rate* est le débit du médium. La portée de cette analyse se limite cependant aux nœuds disposés linéairement.

2.4.2 Métriques de qualité de liens sous 802.11 - LQSR

Les problèmes associés au 802.11 ont mené au développement de différentes métriques servant à qualifier quantitativement la qualité des liens de manière à choisir plus judicieusement un chemin optimal, en allant au-delà du simple critère de minimum de sauts. Draves *et al.* (2004) ont conduit une étude empirique de 3 métriques ETX, per-hop RTT et per-hop packet pair. Ces méthodes sont testées à l'aide d'un protocole basé sur DSR. Toutes ces méthodes se comportent mal en situation de grande mobilité et le critère de sauts minimum demeure la meilleure alternative. Cependant, pour un réseau stable, le critère ETX présente la meilleure performance.

ETX correspond à un compteur de transmissions attendues (expected transmission count). Cette métrique se base sur le taux de perte de paquets entre voisins. Un paquet sonde est envoyé par diffusion générale chaque seconde. Ce paquet inclut le nombre de paquets sondes reçues des voisins dans les dernières 10 secondes. La diffusion générale utilisée par ETX réduit les charges indirectes (overhead) en comparaison avec les méthodes ci-dessous, qui utilisent plusieurs messages à diffusion unique (unicast). De Couto *et al.* (2003) ont décrit ETX comme une métrique incorporant les effets du taux de perte des liens, de l'asymétrie des taux de perte dans les deux directions

d'un lien, et de l'interférence entre les différents liens successifs d'un chemin. L'ETX est calculé à partir de d_f (forward delivery ratio) et d_r (reverse delivery ratio). La probabilité qu'un message soit correctement envoyé et reçu est : d_f * d_r. Le nombre d'essais prévus pour une transmission réussie devient, en considérant chaque essai de transmission comme un essai Bernouilli :

$$ETX = \frac{1}{d_f * d_r} \tag{2.12}$$

Afin d'éviter une synchronisation accidentelle, les paquets sondes sont envoyés avec une variation de +/-0.1 seconde. La métrique ETX d'une route devient la somme des métriques pour chacun des liens la composant. ETX est particulièrement performant pour les petits paquets et ne tient pas compte de la mobilité.

D'un autre côté, la métrique Per-hop RTT calcule le délai d'aller-retour pour un paquet sonde. Ce paquet, envoyé chaque 500ms, contient une estampille temporelle (timestamp). Chaque voisin doit répondre avec un accusé de réception faisant l'écho de l'estampille temporelle. Si les liens sont chargés, les paquets sondes sont mis dans une file d'attente ou retransmis plusieurs fois créant de l'instabilité.

Finalement, la métrique Per-hop Packet Pair Delay calcule le délai entre deux paquets sondes envoyés l'un après l'autre à un voisin. Elle tente de corriger la distorsion de la méthode précédente. Les deux paquets sont envoyés chaque 2 secondes. Le voisin calcule le délai entre ces deux réceptions. Cependant, le travail requis pour le calcul peut être lourd et cette métrique dépend toujours du trafic et cause des interférences.

Le protocole LQSR a été développé afin d'implanter ces métriques. Les tests réalisés avec ce protocole, basé sur DSR, démontrent que les deux dernières métriques présentées y performent mal car elle sont sensibles à la charge de trafic et car, en ajoutant du trafic, elles augmentent les interférences possibles. Le pilote MCL utilisé se situe entre la couche MAC et la couche IP, donc à la couche 2.5. On peut donc utiliser sans modifications IPv4 et IPv6. LQSR utilise une cache de liens, plutôt qu'une cache de routes, ce qui en fait un protocole de routage basé sur l'état des liens, comme OSPF. Lorsqu'une requête de route est reçue, le nœud ajoute son adresse à la route du paquet puis y ajoute aussi la métrique correspondant au lien sur lequel le paquet est reçu, puis le renvoie. Donc, la découverte de route crée une cache de routes mais les liens évoluent avec le temps. Pour les maintenir à jour, le récepteur d'un paquet renverra à la source une réponse de route gratuite afin d'acheminer les nouvelles

métriques. Une seconde méthode proactive de mise à jour consiste en l'envoi d'un message INFO LINK provenant de la source et dans lequel chaque lien est décrit. En cas de bris de liens, la métrique est pénalisée et un message d'erreur de route est acheminé.

Le protocole fut mis à l'épreuve à l'aide d'un réseau simulé composé de 23 nœuds utilisant LQSR. Les résultats démontrent que le débit diminue lorsque le nombre de bonds augmente dû à l'interférence inter bond. De plus, le surdébit dû au trafic de contrôle est apparent pour des routes de 1 bond mais il diminue pour des routes plus longues. On remarque que lorsque les mauvais liens sont éliminés de la cache, les routes tendent à devenir plus longues. Il est important d'envoyer du trafic de fond qui n'utilise pas les métriques de qualité des liens pour le routage afin de repeupler les caches avec des liens potentiellement meilleurs. Il faut trouver un équilibre entre l'ajout et le retrait de liens de la cache pour une opération optimale.

2.5 Répartition de charge de calcul

Répartir la charge de calcul sur divers nœuds est un défi au niveau de l'énergie car la transmission de données est beaucoup plus énergivore que le calcul local. À titre d'exemple, la transmission d'un octet de data pur consomme l'équivalent énergétique de 2800 instructions locales pour le capteur EYES (Nieberg, 2003). Cependant, la répartition garde tout son sens lorsque l'on répartit localement le calcul dans le voisinage du nœud plutôt que de le transmettre jusqu'à un centre de traitement, possiblement très éloigné.

2.5.1 Cohérence des protocoles pour du calcul réparti

Al-Karaki et Kamal (2004) ont présenté les protocoles cohérents qui permettent la coopération entre les nœuds du réseau de capteurs pour le traitement des données. Les protocoles non cohérents, quant à eux, impliquent que les nœuds exécutent des calculs sur les données captées puis envoient l'information traitée vers d'autres nœuds.Ceux-ci manipulent les données à nouveau, dans des tâches d'agrégation, par exemple. Les protocoles cohérents envoient l'information non traitée directement vers des nœuds de calcul. On peut ajouter à l'information une étiquette temporelle mais sans plus. Cette méthode est plus efficace énergétiquement que les méthodes envoyant à un centre

de traitement les données brutes. Les protocoles cohérents requièrent l'envoi d'une quantité importante d'information brute, ce qui nécessite la découverte de chemins optimaux. Le protocole non cohérent débute par la détection d'un évènement, la collecte de données puis le traitement de ces données. Par la suite, afin de partager cette information, il partage son intention à ses voisins. Les topologies partielles sont échangées et on procède à l'élection du nœud qui se chargera des tâches d'agrégation.

Deux algorithmes sont proposés pour choisir le nœud chargé d'accomplir les tâches d'agrégation. Single winner (SWE) choisit un seul nœud. Ce nœud est choisi pour ses réserves énergétiques ainsi que sa capacité de calcul. Un arbre de couverture minimum est ensuite formé. Multiple Winner (MWE) ne se limite pas à un seul nœud choisi. Plusieurs nœuds sont choisis et les nœuds capteurs peuvent envoyer l'information au nœud de leur choix qui lui se chargera de faire parvenir l'information au nœud maître. Ceci réduit le trafic et augmente la vie du réseau. Cependant, l'algorithme est plus lent et les délais et surdébits sont supérieurs.

2.5.2 Approche hybride impliquant l'agent mobile

Xu et Qi (2004) ont présenté les résultats d'une série de tests visant à comparer les performances du paradigme client-serveur et du paradigme agent mobile en vue de développer une approche alliant les avantages de chacun pour un traitement efficace des données. Les tâches des capteurs sont décomposées en trois sections : la lecture de données, le traitement des données et la transmission de ces données. Dans les deux approches, le traitement de données est exactement le même. On s'attarde alors surtout à la lecture de données et la transmission. Les résultats démontrent que pour un grand réseau, la méthode par agents est nettement plus performante. La méthode client-serveur est utile pour des réseaux limités. L'approche par agent mobile utilise un agent qui consiste en un logiciel qui s'auto exécute. Il inclut la méthode qui performe le traitement, mais aussi de l'espace mémoire, un identifiant et un itinéraire de nœuds à visiter. L'agent mobile migre d'un client à l'autre où il s'exécute localement, en utilisant les ressources des clients. L'agent mobile fusionne l'information capturée sur chaque nœud et à partir de cette information partielle, il visite un prochain nœud. Si les données recueillies ont la précision désirée, l'agent peut revenir directement au centre de traitement, sans compléter son itinéraire. Cette approche apporte une solution aux contraintes suivantes du paradigme client-serveur :

- les super-nœuds de traitement requièrent une capacité de traitement et une source d'énergie supérieures ;
- tous les nœuds font parvenir leur information au centre de traitement, ce qui génère un trafic important ;
- les nœuds situés près du centre de traitement sont trop sollicités car tout le trafic doit transiter par eux et ainsi leur réserves d'énergie s'épuisent rapidement.

La méthode par agents est désavantageuse pour des réseaux limités car cette méthode ajoute des charges indirectes dues à la création et à l'envoi des agents mobiles mais aussi dues à l'accès aux fichiers des capteurs. La méthode proposée consiste en un modèle hybride qui se décline sous deux formes. La première déclinaison consiste en créer des grappes et en élisant des têtes de grappes à l'aide d'un protocole comme LEACH. On utilise ainsi peu de grappes incorporant un large nombre de nœuds. On utilise l'approche par agent à l'intérieur de chaque grappe puis l'approche client/serveur entre les têtes de grappes et le centre de traitement. La deuxième déclinaison consiste à nouveau à former des grappes et élire des têtes de grappe avec LEACH mais cette fois, les grappes sont de taille réduite. On utilise alors l'approche agent mobile entre les têtes de grappes et le centre de traitement et l'approche client-serveur au sein de la grappe. Cette méthode sous chaque déclinaison performe mieux en comparaison avec les approches pures de client-serveur ou d'agent mobile.

2.6 Voies de recherche

L'état actuel de la recherche propose une grande richesse de protocoles tant pour les réseaux mobiles conventionnels comme pour les réseaux mobiles de capteurs. Actuellement, le marché commercial présente très peu de solutions mobiles impliquant du routage multi-bonds. Beutel (2005) suggère que c'est dû aux problèmes de l'environnement réel, source d'interférences, de défaillances imprévisibles, d'imperfections et d'interactions humaines. Notamment, cet article suggère que la complexité des algorithmes à développer pour des applications concrètes se doit notamment à cet environnement non déterministe mais aussi au manque d'outils pour la visualisation et le débogage de réseaux de taille importante. Les algorithmes spécialisés pour les réseaux de capteurs se distinguent fondamentalement des algorithmes pour les réseaux mobiles par le fait que la conservation d'énergie devient la préoccupation principale. En effet, on retrouve cette préoccupation tant avec LEACH, qu'avec la Diffusion Dirigée

ou encore avec SPIN. Souvent, les algorithmes ont plusieurs modes de fonctionnement de façon à réduire l'amplitude de leurs tâches lorsque leurs réserves d'énergie décroissent. Les articles scientifiques analysant les réseaux de capteurs tendent de plus en plus vers des réseaux impliquant des centaines ou des milliers de nœuds. Le but visé est d'augmenter la densité de nœuds recouvrant une aire. Cette haute densité a deux objectifs principaux. D'abord, en fusionnant l'information à l'aide de processus collaboratifs, on parvient à réduire le trafic global vers le centre de traitement et à préserver les réserves d'énergie d'un ensemble de nœuds voisins. De plus, on augmente la robustesse et la fiabilité du réseau car même si un nœud a une défaillance, les nœuds voisins prennent la relève et recueillent sensiblement la même information. Ces réseaux de capteurs impliquant une grande quantité de capteurs sont surtout envisagés pour des applications militaires de suivi et de reconnaissance de terrain ou encore pour des situations d'urgence où le déploiement rapide d'un réseau est requis. Il existe cependant certaines applications où le positionnement des nœuds est planifié et où les sources d'énergie ne viendront pas à manquer. On peut penser à des applications où le câblage d'alimentation est installé antérieurement et disponible mais où les infrastructures ne permettent pas l'installation de câblage réseau pour la communication. On peut aussi penser à des applications temporaires où les infrastructures fournissent l'alimentation. Finalement, on peut penser aux applications dont la source d'énergie est renouvelable et n'est donc pas l'enjeu central. Un exemple d'applications est le déploiement d'un réseau sans fil pour un évènement durant lequel on désire recueillir et analyser des données captées par des capteurs : RFID, pression, température, etc. Ces capteurs même avec une source suffisante d'énergie sont tout de même limités dans tous les autres aspects, notamment en capacité de traitement ainsi qu'en mémoire et en fiabilité.

La problématique d'un réseau où chaque nœud fait le traitement des données brutes propres ou provenant d'un nœud voisin est intéressante. En effet, il est possible qu'un nœud détecte un évènement et qu'il veuille agir sur cet évènement. Il est alors possible que les différents capteurs qu'il incorpore surchargent sa capacité de traitement. Il n'est pas toujours recommandable dans ces cas de détourner l'information vers un centre de traitement pour que le centre de traitement lui retourne un diagnostic sur lequel le nœud agira. Il est plus intuitif de traiter l'information localement sur le nœud même ou avec la participation des voisins. La répartition de tâches dans un environnement réparti est un domaine sous étude pour les réseaux

filaires. On tente de répartir les tâches pour exécuter un maximum de tâches dans un temps donné. Becchetti *et al.* (2004) se sont penchés sur l'optimisation de l'étirement moyen d'une tâche, c'est-à-dire la réduction du délai additionnel dans le traitement d'une tâche, délai imputable à la congestion du réseau et des ressources de traitement. Il suggère un algorithme la minimisant, impliquant la théorie des queues. Cet algorithme illustre la puissance de la distribution de tâches sur plusieurs nœuds.

L'effort de cette recherche se concentrera sur le développement d'un protocole qui permettra non seulement d'équilibrer le trafic sur des liens fiables mais qui distribuera aussi la charge de calcul. En effet, il la distribuera sur différents nœuds afin de minimiser le délai additionnel de traitement d'une tâche.

CHAPITRE 3

Protocole de répartition de charge proposé

Dans le chapitre précédent, les solutions actuelles pour le routage des réseaux de capteurs sans fil ont été présentées. Cependant, les protocoles actuels, mus par l'urgence de conserver l'énergie du réseau, ne se penchent pas sur d'autres aspects critiques dudit réseau. En assumant une amélioration de la capacité énergétique des futurs réseaux de capteurs, une des facettes qui doit être considérée est la répartition de travail. Certaines applications, notamment pour la localisation de cibles, ne sont pas en mesure de contrôler la densité de stimuli présents dans un périmètre spécifié. Le traitement des données relatives aux stimuli présents peut devenir trop important dans cette aire du réseau pour les nœuds de traitement qui s'y trouvent. Afin d'alléger leur charge de travail, le protocole de routage peut permettre de distribuer le traitement des données sur des nœuds moins surchargés du réseau. Le protocole présenté dans cette section se charge de la distribution dans le réseau des données pour le traitement relatif à la localisation, mais aussi de l'initialisation et de l'organisation du réseau. Ce chapitre propose une solution au problème de surcharge de données sur les nœuds à l'aide d'un protocole proactif de répartition de charge de calcul et de trafic, nommé RCCT. Après avoir introduit brièvement le protocole RCCT et ses principes de fonctionnement, il présente la motivation, les requis ainsi qu'une liste des hypothèses retenues pour cette étude. Par la suite, le protocole est présenté en détail et les étapes principales sont étudiées individuellement sur le plan fonctionnel ainsi qu'au niveau des structures de données requises. Suivra une analyse mathématique permettant d'exprimer le coût du protocole de façon mathématique.

3.1 Caractéristiques générales du protocole RCCT

Le protocole RCCT est divisé en deux parties. La première partie s'occupe de la gestion de la topologie et la deuxième partie s'occupe du traitement des données capturées.

3.1.1 Gestion de la topologie de la région

Étant donné que le protocole vise la répartition de la charge de trafic et de calcul, il est important de connaître la topologie de la région afin de choisir les chemins optimaux pour le routage. De plus, comme les nœuds de capteurs sont statiques et ont une réserve d'énergie abondante, notre choix s'est porté sur un protocole proactif. Précisément, la connaissance de la topologie se fait à l'aide du protocole OSPF-wireless. Ce protocole (Ahrenholz *et al.*, 2004) est une variante de OSPF v.2 spécifiant le fonctionnement du protocole pour une interface sans fil. À l'aide d'OSPF-wireless, les nœuds obtiennent la table de routage leur permettant de router leurs paquets en utilisant des liens peu utilisés et de meilleure qualité. Les étapes du fonctionnement d'OSPF wireless sont illustrées au tableau 3.1 et seront expliquées de façon plus détaillée ultérieurement.

TABLEAU 3.1 Détermination de la topologie

Étapes	Sous-Étapes	Détails
Découvrir les voisins	À l'aide des HELLO, établir les voisins bidirectionnels à un saut et à deux sauts et choisir les MPR (relais multipoints).	Le voisin à deux sauts est le voisin d'un voisin à un saut du nœud en question. Les MPR sont le groupe minimum de noeuds couvrant les voisins à deux sauts.
Peupler les tables de topologie	À l'aide de LSF envoyés aux voisins directs, échange de LSA. Calcul de la table de routage avec l'algorithme de Dijkstra en utilisant les LSA recueillis.	Un LSA décrit l'état des liens aux nœuds voisins en se basant sur les métriques spécifiées. Le nœud envoie l'annonce aux MPR qui se chargent de le transmettre aux nœuds à deux sauts. L'envoi du LSA se fait par une diffusion générale.

3.1.2 Gestion pour le traitement des données capturées

La gestion pour le traitement des données capturées se fait en deux temps. Dans un premier temps, on établit à l'aide de l'algorithme de d-grappes Max-Min (Amis et Prakash, 2000) les têtes de grappe qui seront responsables du traitement par défaut. Chaque tête de grappe recevra les données capturées par les nœuds dans sa grappe et fera le traitement pour la localisation. Si ses ressources approchent un seuil d'utilisation limite, elle signale des nœuds de la grappe qui seront chargés du traitement de façon temporaire. Elle peut choisir plusieurs chargés de traitement à la fois. La tête de grappe, ou un de ses chargés, effectue alors le traitement et détermine la localisation de la cible. Une fois celle-ci complétée, les résultats sont acheminés à un des nœuds à l'origine de la détection ainsi qu'au serveur ou collecteur de données. Ceci est illustré au tableau 3.2.

3.2 Motivations et requis

Le protocole RCCT est développé spécifiquement pour une application de localisation par TDOA (Time Difference of Arrival). Cependant, son fonctionnement vaut aussi pour les applications réparties où les nœuds doivent analyser de façon collaborative l'information recueillie par les capteurs en vue de prendre une action directe ou d'acheminer les résultats à des collecteurs de données.

Dans quel but développer ces protocoles ? Pourquoi chaque nœud n'analyserait-il que ce dont il est capable ? Qu'arriverait-il si une cible critique n'était pas détectée pour cette raison ? Tout en respectant les requis de l'application présentés plus bas, le protocole RCCT a deux objectifs principaux :

- minimiser la charge de trafic maximal sur les liens du réseau ;
- minimiser la charge maximale de calcul des nœuds du réseau.

Cette optimisation doit se faire sans réduire le nombre d'événements traités par le réseau versus le nombre total d'événements présents. De plus, on doit faire parvenir un nombre maximum de résultats au collecteur, sans en perdre en chemin. On doit également éviter la redondance et éviter de traiter le même événement sur plus d'un nœud. Tout cela devrait conduire à traiter un maximum d'événements (ou la totalité). En vue d'atteindre de tels objectifs, il serait opportun de tenter de maintenir à un

TABLEAU 3.2 Traitement des nœuds

Étapes	Sous-Étapes	Détails
Formation des grappes	Choix des têtes de grappe	Heuristique Max-Min pour le choix des têtes de grappe de la région
	Association des membres à une tête de grappe	Les nœuds s'associent à une tête de grappe ou sont sommés de joindre une tête particulière. Découverte des passerelles, nœuds situés en bordure de la grappe
Traitement d'un événement (ou stimuli)	Calculs entamés par la tête de grappe	La tête de grappe est le responsable du traitement pour sa grappe, tant que ses ressources (capacité de calcul) le lui permettent
	Choix du chargé de traitement	Si le nœud responsable est trop occupé pour s'occuper de ce nœud, il peut choisir, à l'aide de ses tables de capacités de nœuds, le nœud le plus approprié pour le traitement. Il publie son identité et les nœuds de la grappe lui envoient directement l'information. Sur réception, celui-ci calcule la localisation des cibles.
	Acheminement des résultats au collecteur	Une fois le traitement terminé, les résultats sont acheminés au collecteur de données ou à un nœud à l'origine de la détection. Ce message est acheminé à l'aide des tables de routage.

minimum le trafic de contrôle requis pour la gestion du réseau comme pour la gestion des événements.

En plus de chercher à satisfaire les objectifs cités plus haut, le protocole doit également répondre à certains requis fondamentaux dans le contexte de l'application de localisation de cibles. Notamment, un des requis consiste à déterminer la position à l'aide des données recueillies par un minimum de trois nœuds distincts, un processus nommé triangulation. Chaque capteur reçoit une séquence provenant des cibles qui inclut un code identifiant uniquement la cible repérée ainsi qu'une estampille temporelle. De plus, les nœuds de capteurs seront statiques mais les cibles pourront bouger.

Finalement, cette application est prévue pour l'Ethernet sans fil sous la spécification pour les couches liaison et physique du standard 802.11.

Afin de cerner le sujet de cette étude, certaines hypothèses sont adoptées. D'abord, il est assumé que le réseau est indépendant et n'est rattaché à aucun autre réseau avec lequel il doit communiquer. Ceci se fait dans le but de limiter les communications et le routage au niveau inter-réseau. Le centre de traitement peut à son tour communiquer avec d'autres réseaux au besoin. Aussi, le réseau ne fonctionne que sur un type d'interface réseau, c'est-à-dire le 802.11. De ce fait, la solution proposée se situe à la couche de routage, soit la couche trois du modèle OSI, et n'a pas à gérer des éléments des couches un et deux. De plus, le protocole est prévu pour une disposition uniforme des nœuds dans la région, comme une grille.

Le modèle dépend de plusieurs facteurs. La liste non exhaustive suivante présente les paramètres les plus susceptibles d'influencer le bon fonctionnement du protocole RCCT :

- unités de temps requises pour traitement d'un événement relié à une cible ou envoi d'un paquet ;
- longueur des queues de transmission des nœuds du réseau ;
- taux de perte de paquets sur le réseau, i.e. la qualité des liens ;
- nombre de nœuds ;
- nombre de cibles ;
- vitesses des cibles ;
- disposition des nœuds ;
- énergie disponible dans le réseau ;
- énergie requise pour chacune des tâches : traitement d'événements, envoi de paquets ;
- capacités des liens ;
- capacités des unités en termes de traitement ;
- liens unidirectionnels/bidirectionnels.

Les facteurs principaux sur lesquels le protocole s'attardera sont la distribution des cibles ainsi que les capacités de traitement des nœuds et de trafic des liens.

3.3 Description du protocole RCCT

Cette section présente de façon plus précise chacune des étapes formant le protocole. Elle met à jour les motivations entraînant l'inclusion d'une étape et expose le détail de ces étapes en fournissant des informations sur les structures de données et paquets requis.

3.3.1 Gestion de la topologie

La découverte et gestion de la topologie se fait par l'algorithme OSPF wireless. Cet algorithme est constitué de deux parties : la découverte des voisins, et le routage et l'état des liens.

Découverte des Voisins

Le protocole débute par la découverte des voisins. Cette découverte se fait à l'aide de messages HELLO envoyés par diffusion générale. Chaque nœud y publie ses voisins symétriques, asymétriques et les relais multipoints. Les relais multipoints sont des voisins directs symétriques choisis par chaque nœud pour transmettre leurs annonces d'états de liens (LSA) un bond plus loin. Le nœud les calcule à l'aide d'une heuristique générant le groupe minimum de nœuds permettant d'atteindre tous les voisins symétriques situés à deux sauts de distance du nœud calculateur.

La Figure 3.1 illustre comment se fait le choix du relais multipoints. Il est à noter que le relais multipoints N7 est le premier choisi car il est le seul voisin direct de N1 permettant d'accéder au voisin symétrique à deux bonds de distance N8. Il faut noter aussi que le nœud N5 n'est pas choisi car le nœud N9 n'a qu'une connectivité asymétrique. Finalement, N3 et N4 sont également choisis car ce sont les nœuds permettant de couvrir le plus efficacement les voisins à deux bonds de distance non couverts restants.

Routage et états des liens

Les annonces d'états des liens, ou LSA, sont des paquets envoyés périodiquement et décrivant l'état des liens avec chacun des voisins à partir des métriques choisies. Ces métriques quantifient le potentiel d'un lien à l'aide de deux critères : la qualité d'un lien ou sa capacité de transmission sans erreurs, et la disponibilité du lien ou

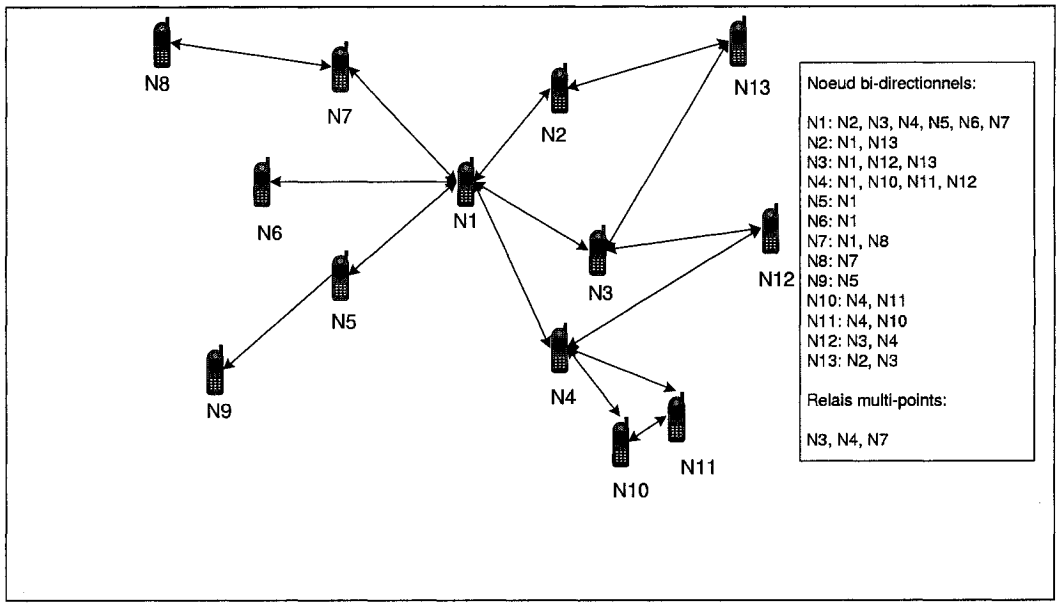

FIGURE 3.1 Choix des mpr

le trafic sur ce lien par rapport à sa capacité maximale. Une autre métrique, *quantitéTraitement*, est également transmise par le LSA. Cette métrique sert toutefois à la sélection du chargé de traitement plutôt qu'au calcul de la table de routage, et sera décrite plus tard. Chaque voisin aura en sa possession des LSA provenant de chacun des nœuds de la région et les utilisera pour calculer, à l'aide de l'algorithme de Dijkstra, se table de routage.

Métriques pour le calcul de la table de routage

Les deux métriques choisies pour qualifier le lien sont complémentaires. La métrique *quantitéLien* décrivant le trafic sur tous les liens sortant du nœud calculateur se définit comme suit :

$$quantitéLien = \text{espace utilisé sur file d'attente TX/taille de file d'attente TX,}$$

où les quantités sont mesurées en octets. La métrique *qualitéLien* se mesure de la façon suivante :

$$qualitéLien = ((\text{sondes reçues TX/sondes envoyées RX}) + (\text{sondes reçues RX/sondes envoyées TX}))/2,$$

où on mesure la proportion des sondes reçues correctement entre deux nœuds, dans les deux directions possibles de trafic. Les métriques sont complémentaires car *quantitéLien* décrit l'état du trafic sortant du nœud, trafic dû à des paquets passant en transit par le nœud ou encore générés suite à un traitement sur ce nœud. La métrique *qualitéLien* décrit une situation anormale. Par exemple, si le lien se perturbe suite à l'introduction d'un obstacle, *qualitéLien* rapportera l'anomalie et évitera qu'on retransmette sur ce lien des paquets qui ne parviendront pas à leur destination. Chacune de ces métriques se situe entre 0 et 1. Afin de maximiser les contributions de chacune des métriques, elles sont combinées pour former la métrique globale suivante :

$$\text{métriqueGlobale} = 1 / (\textit{quantitéLien} * \textit{qualitéLien}),$$

donc, plus un lien sera inadéquat, plus ce lien aura une métrique globale élevée. Cette métrique globale correspond au coût d'un lien selon l'algorithme de Dijkstra. Ainsi, lors du calcul des chemins optimaux, on valorisera les liens ayant une métrique globale faible, c'est-à-dire une qualité de lien et une capacité pour du trafic additionnel supérieures.

Calcul de la table de routage

La table de routage permet de choisir le prochain nœud auquel envoyer un message destiné à un nœud précis. Ce nœud est choisi en fonction des métriques favorables caractérisant le lien l'unissant au nœud émetteur. Toutefois, la table de routage est appelée à changer car les caractéristiques des liens varient avec le temps, dû à l'usage qu'on fait des nœuds et à des événements perturbateurs. Cependant, une faible variation dans les métriques ne doit pas engendrer le recalcul immédiat de la table de routage pour éviter de recalculer trop fréquemment. Des paliers seront établis : EXCELLENT, MOYEN, MAUVAIS et NUL, correspondant à des intervalles de valeurs de la métrique globale. Le recalcul de la table de routage ne se fera que si un passage d'un palier à un autre se fait et si la différence entre la valeur dans le palier précédent et dans le palier actuel dépasse une valeur minimale établie. Ainsi, on évite que les fluctuations entre deux paliers ne génèrent des calculs répétés de la table de routage.

Arbre de recouvrement minimum

Les messages LSA sont gardés dans une structure de données. Un graphe de connectivité est formé à partir des LSA. On peut créer un arbre de couverture minimum par la suite en appliquant l'algorithme de Dijkstra à l'aide des trois métriques choisies. C'est de cet arbre de couverture minimum que sera finalement extraite la table de routage.

3.3.2 Gestion du traitement des données capturées

Tel que spécifié dans la section des requis, la localisation d'une cible peut être déterminée lorsque les données d'au moins trois nœuds sont utilisées pour son calcul. Chaque nœud captant les données d'une cible peut potentiellement devenir le responsable du calcul de la position de cette cible. Cependant, le choix dynamique de ce responsable entraîne une quantité de trafic de contrôle importante sur le réseau. Si le nombre de cibles devient élevé, ce trafic de contrôle deviendra rapidement trop important, et nuira à la transmission de données utiles de position. À cette fin, on établit des grappes et on élit des nœuds à la tête de ces grappes. Ces têtes de grappe sont chargées de recevoir ou de gérer le traitement de toutes les données recueillies par des membres de leur grappe.

Formation des grappes

L'algorithme choisi pour la formation des grappes est l'algorithme Max-Min pour la formation de grappes de d bonds, algorithme proposé par Amis et Prakash (2000). Cet algorithme permet de former des grappes où les nœuds se trouvent à une distance maximale de d bonds de la tête de grappe. Par contre, il ne garantit pas un nombre précis de nœuds par grappe ni un nombre minimum de nœuds à la jonction de cette grappe avec les grappes voisines. L'algorithme intègre trois étapes : la diffusion maximum (*floodmax*), la diffusion minimum (*floodmin*), et la convergence (*convergecast*).

Diffusion maximum

Le processus de sélection des têtes de grappe se base sur la valeur du numéro d'identification (*id*) des nœuds. Durant la première étape de *floodmax*, d rondes d'envoi

par diffusion générale du *id* sont requises. À chaque ronde, le nœud doit choisir le *id* dont la valeur est supérieure et en faire le sien pour la prochaine ronde. Ainsi, les *id* aux valeurs les plus hautes sont propagés à travers le réseau sur d sauts. Les *id* retenus par les nœuds à la fin de chaque ronde sont gardés en mémoire et ceux retenus à la dernière ronde constituent les futures têtes de grappes.

Diffusion minimum

Après *floodmax*, les grappes aux têtes possédant un *id* très élevé auront plus de nœuds que les grappes aux têtes ayant un *id* inférieur. L'étape de *floodmin* a pour but de corriger cette situation. Cette étape consiste en d rondes également, où le *id* le plus petit est adopté par chaque nœud à chaque ronde.

Concept de paire

Le *id* choisi à chaque ronde de l'étape de *floodmax* et de *floodmin* est gardé en mémoire. Lorsqu'un *id* apparaît tant durant les rondes *floodmax* comme de *floodmin*, il constitue une paire. Pour chaque nœud, la tête de grappe qu'il choisit est le nœud correspondant au *id* le plus petit ayant formé une paire.

Convergence

Les nœuds doivent ensuite informer la tête de grappe de leur choix. Les nœuds limitrophes sont choisis pour initier ces messages de convergence vers la tête de grappe. Pour déterminer qu'un nœud est limitrophe, donc situé en bordure de deux grappes, tous les nœuds envoient par diffusion générale un message annonçant à quelle grappe ils appartiennent. Si les messages reçus de ses voisins indiquent que les voisins appartiennent à une autre grappe, le nœud est déclaré limitrophe. Il initie alors un message vers la tête de grappe. Tous les nœuds intermédiaires recevant le message ajoutent leur *id* également au message envoyé et le renvoie vers la tête de grappe. Ainsi, lorsque la tête de grappe reçoit ces messages de convergence, elle connaît immédiatement les membres de sa grappe.

Exceptions durant la formation des grappes

Certaines situations particulières font exception au fonctionnement régulier de cet algorithme Max-Min. Notamment, si une cible obtient son propre *id* dans une

des rondes de l'étape *floodmin*, mais il n'apparaît pas durant les rondes *floodmax*, ce nœud deviendra tête de grappe. Aussi, si aucune paire n'est formée suite à *floodmax* et *floodmin*, le *id* gagnant suite aux rondes de *floodmax* sera choisi tête de grappe. Finalement, durant l'étape de convergence, si une tête de grappe est sur le chemin entre un nœud et la tête de grappe qu'il a choisi, la tête de grappe intermédiaire s'appropriera ce nœud et l'informera de la nouvelle grappe à laquelle il appartient maintenant. La Figure 3.2 démontre ces exceptions. Par exemple, le nœud 28 choisit la tête de grappe 100 mais le message d'association passe par le nœud 73. Le nœud 73, étant une tête de grappe, assimile le nœud 28 dans sa grappe. Un autre exemple est le nœud 73 qui, quant à lui, s'est proclamé tête de grappe lorsqu'il s'est vu dans la table de résultats de *floodmin*, bien qu'il n'apparaisse pas dans la table de résultats *floodmax*.

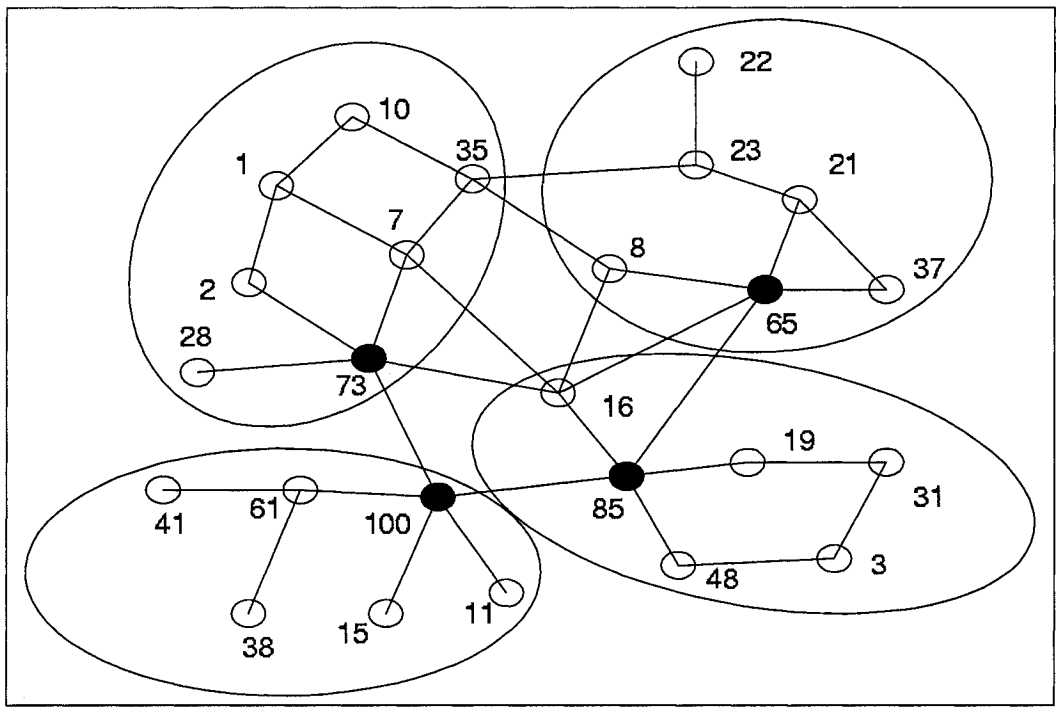

FIGURE 3.2 Choix des grappes

Responsable de traitement

La tête de grappe devient le centre de traitement pour toutes les captures faites sur le territoire de la grappe. Ainsi, tant que sa capacité de traitement est sous le seuil limite déterminé, et qu'elle dispose donc de ressources suffisantes pour accomplir les traitements pour la localisation, elle se charge de toutes les détections dans sa grappe. Une fois la localisation complétée, elle transmet le résultat à un élément à l'origine des captures ou au centre de traitement.

Tête de grappe de rechange

La tête de grappe a un rôle très important. Afin de s'assurer de son bon fonctionnement, elle doit envoyer à intervalles réguliers, un message LIFE où elle confirme son rôle de tête de grappe et annonce la tête de grappe de rechange, au cas où surviendrait une défaillance de la tête de grappe. Le message LIFE indique également un chargé de traitement temporaire au besoin.

Capacité de traitement des nœuds et chargé de traitement

La métrique *quantitéTraitement*, transmise par les LSA, n'est pas utilisée pour calculer la table de routage. Cette métrique devient plutôt utile pour répartir le traitement des cibles. La métrique se définit comme suit :

quantitéTraitement	=	nombre de cibles à traiter/nombre maximum de cibles possibles de traiter en une seconde.

Lorsque le responsable de traitement observe que sa métrique *quantitéTraitement* respective a atteint le seuil limite, il inspecte le champ *quantitéTraitement* dans la table de routage pour les nœuds de sa grappe et choisit un nœud peu utilisé qui deviendra le prochain chargé de traitement. Lors du prochain message LIFE, ce chargé de traitement sera publicisé et les nœuds de la grappe pourront lui envoyer leurs données pour le traitement. Il faut bien sûr choisir la taille des grappes en fonction du nombre de cibles maximum prévu et de leur proximité. Plus la densité de cibles augmente, plus il convient de créer des grappes petites afin de multiplier le nombre de nœuds traitants à un même moment dans le réseau. S'il y a beaucoup de cibles, chaque chargé de traitement sera en fonction pour un temps plus court, temps durant

lequel il atteindra son seuil maximum de cibles à traiter. Il faut cependant compter un certain temps afin d'aviser les membres de la grappe du choix d'un nouveau chargé de traitement. La Figure 3.3 illustre l'élection du chargé de traitement. On peut y voir que le chargé n'est choisi que parce que le responsable de traitement est trop chargé, avec un niveau au dessus du seuil limite de traitement. Pour cette raison, il choisit pour chargé de traitement le nœud N6 qui offre le niveau le plus bas en termes de ressources utilisées.

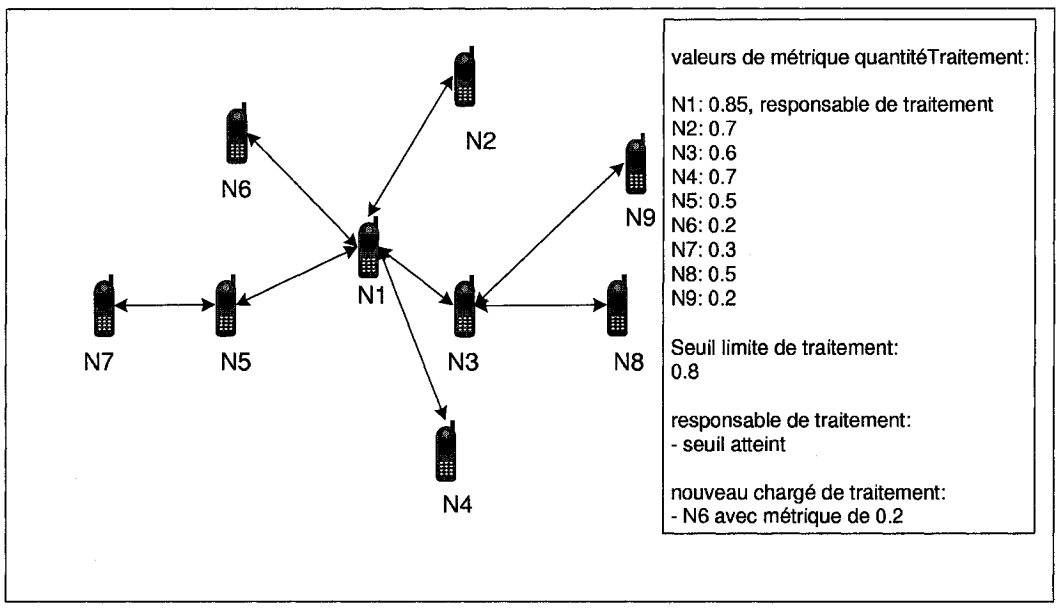

FIGURE 3.3 Choix du chargé de traitement

Congestion et nœuds multiples de traitement au sein d'une grappe

Il peut arriver qu'il y ait une telle densité de cibles dans une grappe que le traitement sur un seul nœud, que ce soit par le responsable de traitement (tête de grappe) ou par un seul chargé de traitement, ne suffise plus. Le responsable de traitement peut alors choisir d'avoir recours à plusieurs chargés de traitement. En effet, les chargés sont des nœuds, choisis par la tête de grappe et activés sur réception d'un message LIFE, qui envoient par diffusion générale les valeurs des cibles qu'ils détectent. Tout nœud situé à proximité et ayant détecté cette même cible envoie alors au chargé ses données, plutôt qu'à la tête de grappe. Ainsi, toutes les cibles détectées par les chargés sont traitées localement. Le voisin d'un chargé lui envoie les données concernant les

cibles détectées conjointement et envoie les données concernant les autres cibles au chargé de traitement. Cette méthode a pour avantage de multiplier le nombre de nœuds traitant au sein d'une grappe. Lorsque la densité de cibles diminue, la tête de grappe enverra un nouveau message LIFE dans lequel elle reprendra graduellement le contrôle du total des cibles sur le territoire recouvert par les membres de sa grappe. De plus, si le chargé devient surchargé lui-même, il peut réduire le nombre de cibles à traiter en envoyant certaines cibles directement à la tête de grappe plutôt qu'en les diffusant à tous pour les traiter par la suite. De plus, si deux chargés sont voisins, et qu'ils diffusent un message concernant la même cible, leurs voisins ayant également détecté cette cible enverront leurs données au nœud ayant le *id* le plus élevé. La Figure 3.4 illustre le fonctionnement des chargés de traitement. Le nœud chargé N4 diffuse ses données pour la cible N1. Les nœuds N3 et N5 lui envoient alors leurs données pour cette cible. Toutefois, le nœud N3 choisit d'envoyer ses données pour la cible 2 à la tête de grappe car il n'a pas reçu de message au sujet de cette cible du nœud N5. Les nœuds N6 et N7 choisissent également d'envoyer leurs données à la tête de grappe cr ils ne sont pas à proximité d'un chargé.

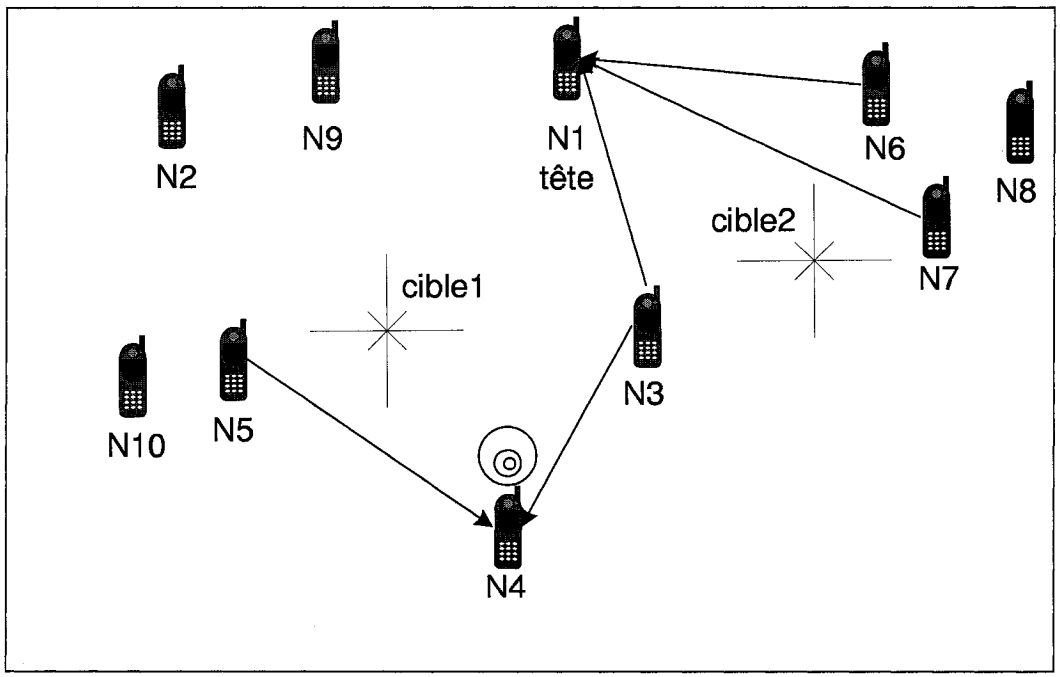

FIGURE 3.4 Fonctionnement des chargés

Nœuds limitrophes

Les nœuds limitrophes, ou passerelles, situés en bordure d'une autre grappe, agissent différemment des nœuds situés à l'intérieur d'une grappe. Si trois nœuds limitrophes appartenant à trois grappes différentes n'envoient leur données pour une cible qu'à leur tête de grappe, la cible ne sera pas détectée par manque de partage d'informations entre les grappes. Ainsi, les nœuds limitrophes, situés en bordure de la grappe, ne transmettent pas directement au chargé de traitement mais plutôt diffusent leurs données à tous, Ainsi, les nœuds d'une autre grappe peuvent les récupérer et les envoyer à leur tête de grappe. De cette façon, une cible située à mi-chemin entre deux grappes ne passera pas inaperçue par manque de données dans chacune des grappes. La Figure 3.5 en est un exemple. On voit que des nœuds limitrophes détectent une même cible. Se sachant nœuds limitrophes, ils envoient les données lues par diffusion générale. Le nœud ayant le plus haut *id*, le N5, récupère les données des nœuds N3 et N4 et les envoie alors à sa tête de grappe, le N6.

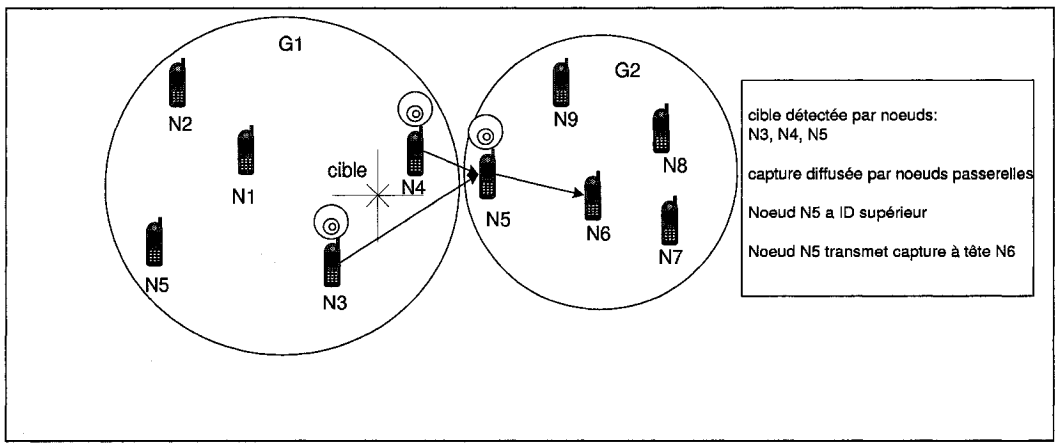

FIGURE 3.5 Traitement et nœuds limitrophes

Discussion sur le traitement par têtes de grappe

Cette méthode offre certains avantages par rapport à une méthode prônant l'envoi systématique des données vers le centre de traitement :

- Les données capturées pour une cible occasionnent plus de paquets et de taille plus importante que les paquets contenant un résultat de localisation. S'ils

doivent traverser le réseau pour atteindre le centre de traitement, ils engendreront un trafic important. Les têtes de grappe permettent un traitement plus localisé et limitent ainsi le trafic.

- Si un nœud à l'origine de la détection doit agir suite la détection, notamment en déclanchant une alarme, il devient encore plus critique d'effectuer le traitement près du nœud d'origine. L'envoi des données au centre de traitement peut être coûteux ; il devient plus onéreux si le résultat de ces calculs doit être réacheminé jusqu'à sa source.
- Le responsable de traitement connaît la topologie du réseau suite à l'application du protocole OSPF-wireless. Il est en mesure de rediriger le trafic vers le nœud le plus avantageux, dans sa grappe ou pas, afin que ce nouveau chargé de traitement puisse soulager sa grappe en offrant des ressources abondantes pour le traitement.
- En choisissant la tête de grappe comme nœud de traitement par défaut, le trafic de contrôle est réduit à un minimum. En effet, les nœuds peuvent envoyer automatiquement leurs données captées à ce nœud. S'il devient surchargé, il dispose des outils nécessaires, à l'aide des métriques de performance du réseau, afin de choisir le chargé de traitement qui prendra la relève. Il est ainsi l'unique décideur et prend la décision sans devoir établir d'échanges avec les membres de sa grappe.
- De plus, l'ambigüité est à un minimum car le responsable de traitement est le nœud traitant. Si ce n'est pas le cas, il avise les membres de sa grappe de l'identité du chargé de traitement à l'aide de messages LIFE.
- En étant assigné à une grappe, un nœud situé à mi-chemin entre deux nœuds traitants sait clairement à quel nœud traitant envoyer ses données : la tête de sa grappe.

Certains aspects de cette solution doivent toutefois être surveillés pour ne pas affecter la performance du protocole, notamment :

- Le volume de trafic destiné au responsable de traitement et ses voisins doit être tenu sous un seuil maximum car tout le trafic a pour destination ou origine ce nœud ; le responsable de traitement peut toutefois rediriger le trafic vers un chargé de traitement si ces liens sont surchargés.
- La multiple détection d'une cible doit être évitée lorsque celle-ci est située entre deux grappes ; les nœuds limitrophes facilitent la multiple détection en diffu-

sant leurs valeurs lues. Cependant, cette méthode évite que la cible passe inaperçue. Ce problème peut être évité si seul le nœud limitrophe ayant le *id* le plus haut peut retransmettre les données des nœuds limitrophes voisins à sa tête de grappe.

- Une variation de métriques trop fréquente engendre de multiples calculs des tables de routage. Il faut établir des paliers permettant un recalcul de la table de routage seulement lorsqu'il est nécessaire pour refléter un changement important dans l'état des liens d'un nœud.
- Le nombre de cibles à traiter peut croître par manque de données complètes pour le traitement. Si le chargé de traitement ne reçoit pas de nouvelles lectures pour une cible donnée à un temps donné, il peut enlever cette cible de sa table de traitement après un intervalle de temps spécifié. Le chargé de traitement conserve les informations relatives à une cible ainsi que l'identifiant du nœud à l'origine de sa détection pour transmettre à ce nœud la position de localisation calculée. Les nœuds limitrophes maintiennent les informations relatives a une détection durant un intervalle de temps donné afin d'identifier les paquets relatifs à cette cible envoyés par des nœuds limitrophes voisins. Ils peuvent ainsi choisir de les retransmettre à leur tête de traitement.

3.4 Analyse du protocole

Dans cette section, nous effectuons une analyse théorique du comportement de notre protocole. De manière plus spécifique, nous analyserons d'une part les messages de contrôle et l'espace requis pour le protocole OSPF-wireless, et d'autre part les messages et l'espace requis pour la formation des grappes.

3.4.1 Messages de contrôle requis pour le protocole OSPF-wireless

Dans une première étape, nous établirons les coûts en termes de messages pour l'opération du protocole proactif OSPF-wireless. Le protocole intègre les étapes suivantes :

- envoi de messages HELLO périodiques ;
- envoi de messages LSF périodiques ;

- renvoi de messages LSF pour un voisin symétrique lorsque choisi comme relais multipoint pour ce nœud.

Si f_h est le nombre de messages HELLO par seconde et f_l le nombre de messages LSF par seconde, le coût en messages est le suivant pour l'opération du protocole OSPF-wireless :

$$\sharp \text{ messages(OSPF)} = f_h + f_l + m*f_l,$$

où m est le nombre de voisins directs du nœud ayant choisi ce nœud comme relais multipoints. Ainsi, le nœud enverra non seulement les messages HELLO et LSF périodiquement selon un intervalle donné (une valeur de dix secondes est recommandé), mais il doit retransmettre au même intervalle les messages LSF reçus des voisins directs symétriques l'ayant choisi comme relais multipoints pour atteindre leur voisins symétriques situés à une distance de deux bonds.

3.4.2 Espace requis pour le protocole OSPF-wireless

Le but du protocole OSPF-wireless est de générer une table de routage incluant une entrée pour tous les éléments de la région. Pour parvenir à ce but, deux étapes de calcul sont importantes. D'abord, il faut calculer les relais multipoints à partir des tables de voisins directs et de voisins symétriques à deux bonds de distance. Puis, à partir des LSA extraits des messages LSF, il faut calculer l'arbre de recouvrement minimum permettant de générer la table de routage.

Définissons certaines variables pour le réseau :

N : nombre de nœuds dans le réseau
I : nombre de voisins symétriques à un bond de distance
J : nombre de voisins asymétriques à un bond de distance

Ainsi, les coûts en mémoire pour la gestion du voisinage sont les suivants :

Table de voisins	=	I + J entrées
Table de voisins à deux bonds de distance	=	I * I entrées
Table de relais multipoints	=	I (maximum) entrées

En effet, le nœud aura une entrée pour chacun de ses voisins directs, qu'ils soient connectés de façon symétrique ou asymétrique. De plus, il conserve une entrée pour chacun des voisins symétriques de ses voisins symétriques directs. Cette table permet

de choisir les relais multipoints qui sont formés, dans le pire des cas, par tous les voisins symétriques du nœud mais qui, généralement, sont formés par une portion des voisins symétriques.

Les coûts en mémoire pour la gestion des liens et la table de routage sont les suivants :

Table de LSF = Table de LSA = Arbre de recouvrement minimum = Table de routage = N entrées

où N est le nombre de noeuds dans le réseau. En effet, lorsqu'un nouvel LSF ou LSA parvient à un nœud, il remplace l'ancien message. À partir des LSA, l'arbre de recouvrement minimum est calculé à l'aide de l'algorithme de Dijkstra. Ensuite, la table de routage est formée à partir de l'arbre généré.

3.4.3 Messages requis pour la formation des grappes

Les messages requis pour la formation des grappes dépendent exclusivement du nombre de sauts séparant la tête de grappe de ses membres les plus éloignés. Voici les coûts en messages requis par nœud pour la formation des grappes où le nombre de sauts est d, le nombre de nœuds passerelles est k, et le nombre de nœuds membres de la grappe est p :

floodmax	$=$	d
floodmin	$=$	d
convergecast	$<$	$k+1$

L'étape *floodmax* pour l'élection des têtes de grappe requière d messages, puis l'étape *floodmin* pour balancer les nœuds dans les grappes requière d messages. L'étape *convergecast* requière les p nœuds membres de la grappe de diffuser leur grappe d'appartenance pour déterminer s'ils sont des nœuds limitrophes. Puis, les k nœuds passerelles seront à l'origine d'un message vers la tête l'informant des nœuds qui composent la grappe. Chaque nœud membre interne recevra au moins un de ces messages auquel il ajoutera son identifiant pour le renvoyer vers la tête de grappe.

3.4.4 Espace requis pour le choix de têtes de grappe

L'espace requis pour déterminer les têtes de grappe est le suivant :

floodmax	=	d entrées
floodmin	=	d entrées
paires	<	d entrées

L'algorithme requière $2d$ entrées pour garder les gagnants des rondes de *floodmax* et *floodmin* et un maximum de d entrées pour garder les nœuds correspondant à des entrée répétée aux étapes de *floodmax* et *floodmin*, désignés par l'expression "paire".

3.4.5 Taille des liens et messages

L'envoi des messages sur les liens diminue la capacité disponible de ceux-ci. Les têtes de grappe ou tout autre chargé de traitement doivent prendre garde de saturer les liens leur permettant de communiquer avec les membres de leur grappe. Mais quel est le nombre de paquets maximum que ces nœuds peuvent supporter sur un lien ?

Le modèle utilisé aux couches MAC et PHY est le 802.11G. Il est possible de calculer approximativement le nombre maximum de paquets que le lien peut accepter durant une seconde. Le 802.11G est caractérisé par un débit maximal de 54 Mbps. Bien que les paquets utilisés par le protocole RCCT dépasse rarement les 32 octets, il faut aussi calculer les entêtes IP, MAC et PHY. L'entête IP est de 20 octets, celle de la couche MAC est de 30 octets, 38 octets avec le LLC et SNAP et celle de la couche PHY de 24 octets. Ainsi, un paquet typique aura une taille de 114 octets. Ceci permettrait un débit d'environ 60.000 paquets.

3.5 Synthèse du protocole

Le protocole RCCT est un protocole qui se penche sur la problématique de la répartition de tâches à l'intérieur d'un réseau de capteurs sans fil. Le protocole utilise diverses approches afin de tirer le maximum des ressources disponibles sur le réseau. Notamment, il utilise le protocole OSPF wireless avec des métriques évaluant la qualité et la disponibilité des liens. Ces métriques permettent de choisir des chemins moins utilisés et augmentent les chances de faire parvenir avec succès un message à sa destination.

Le protocole RCCT met aussi l'accent sur le choix du responsable de traitement. Le choix de l'algorithme Max-Min pour la formation de grappes à d bonds permet de créer une certaine hiérarchie et centralisation de la prise de décisions, tout en res-

tant à proximité de l'endroit de détection. Les têtes de grappe ainsi choisies sont non seulement le responsable de traitement lorsque la densité de cibles est basse, mais aussi sont en charge de déterminer les chargés de traitement lorsque cette densité augmente. Les chargés de traitement sont choisis pour leur grande capacité de traitement à un instant précis et ont pour but de soulager la tête de grappe des tâches de traitement. Ils permettent un traitement localisé et ne servent qu'à traiter les données pour lesquelles ils sont à l'origine. Ainsi, le protocole RCCT se trouve à reléguer le centre de traitement à des tâches secondaires telles le suivi de l'évolution des cibles et la génération de statistiques. En permettant que les nœuds traitent les données localement, le protocole permet une plus grande réactivité. Ainsi, les nœuds disposent immédiatement des résultats de détection. Ceci permet une prise de décision se suivant d'une action, telle, par exemple, le déclenchement d'une alarme. Cette organisation permet une relative indépendance par rapport au centre de traitement.

CHAPITRE 4

Implémentation et résultats

Le problème considéré est un problème d'optimisation. En effet, dans un réseau de capteurs sans fil donné, on souhaite réduire à un minimum le temps requis de traitement pour la localisation de cibles tout en maximisant le nombre de cibles détectées. La principale difficulté est qu'afin de localiser une cible, les données recueillies pour cette cible par trois nœuds du réseau doivent être combinées afin de déterminer sa position. Cela engendre une communication entre les nœuds pour partager les données recueillies au sujet des cibles. De plus, le nœud choisi pour le calcul est plus sollicité que ses voisins en raison de cette charge de calcul et du trafic plus important lui étant adressé. Dans le chapitre précédent, la solution théorique proposée pour alléger le trafic et la charge de calcul sur les nœuds a été présentée. Le chapitre présent se centrera plus sur le plan d'expérimentation devisé pour démontrer la validité de la solution. L'expérience a pour but de créer une simulation qui, tout en reprenant les éléments critiques et plus représentatifs d'un scénario réel, fait abstraction des éléments superflus afin de démontrer l'influence des critères observés. Cette section mettra donc de l'avant d'abord les objectifs de l'expérience et les hypothèses quant à l'impact prévu de la méthode sur ces objectifs, puis présentera la méthode utilisée en définissant son fonctionnement, les variables et métriques de contrôle, l'environnement de test et les scénarios investigués. Ensuite, les résultats obtenus seront présentés et analysés afin d'en ressortir les principales tendances tout en soulignant les difficultés rencontrées et leur impact sur la solution finale.

4.1 Objectifs expérimentaux

Suite à la solution présentée, l'implémentation se veut surtout une validation de celle-ci. La solution amène deux éléments principaux : une division des nœuds en grappes, avec une tête de grappe qui coordonne ses sujets, et avec divers mécanismes au sein de la grappe pour assurer son efficacité de traitement, et un protocole de rou-

tage pour l'ensemble du réseau basé sur des métriques de performance liées au trafic et à la charge de calcul. L'expérience pourra démontrer la valeur de cette approche en surpassant des approches alternatives. Notamment, voici des objectifs adressant les aspects prédominants de la méthode proposée :

- Démontrer que l'approche par protocole proactif du protocole RCCT est plus intéressante que l'approche par protocole réactif.
  L'alternative consiste à utiliser AODV pour acheminer les paquets au centre de traitement. On pourra évaluer la performance des deux méthodes en calculant le nombre total moyen de paquets requis pour la localisation, le nombre moyen de paquets de contrôle requis pour une localisation ainsi que le délai moyen requis pour une localisation. Pour obtenir une image contrastée, on pourra varier le nombre de nombre de cibles dans le réseau.
- Démontrer l'influence des paramètres de l'algorithme Max-Min pour la formation de grappes sur la performance globale, notamment en variant le nombre de sauts de distance entre les membres de la grappe et leur tête.
  On pourra comparer la taille des grappes en fonction du nombre de cibles localisées versus le nombre total de cibles.
- Démontrer l'influence de la métrique de liens utilisée dans le protocole RCCT sur la performance globale.
  On pourra comparer la performance de localisation pour une cible en utilisant cette métrique plutôt qu'en octroyant une valeur unitaire à tous les liens, ce qui indique que tous les liens sont équivalents et qui conduit à choisir le chemin au nombre de sauts minimum. Cette comparaison se fera en variant le nombre de cibles.
- Démontrer l'influence du processus de choix de chargés de traitement et voir si l'ajout de chargés de traitement améliore les localisations de cibles pour un nombre de cibles variées dans le réseau.

4.2 Hypothèses d'expérience

La méthode choisie nous permet d'anticiper un certain comportement du système évalué. Quelques hypothèses sur les résultats peuvent être formulées.

4.2.1 Première hypothèse

Le protocole proactif requière plus de messages de contrôle pour mettre en place les tables de routage par rapport à un protocole réactif. Il permet cependant de faire des choix judicieux de routage en sachant quels nœuds sont moins utilisés. De plus, il peut ainsi connaître également les capacités de traitement des nœuds du réseau, plus difficiles de découvrir dans un réseau basé sur un protocole réactif. Il faudra toutefois confirmer la validité de ce choix par des simulations.

4.2.2 Deuxième hypothèse

L'utilisation de la formation de grappes permet une gestion centralisée mais localement. On s'attend donc a plus de trafic sur les têtes de grappe et un plus haut taux de perte de paquets dans leur proximité mais aussi un temps de traitement plus rapide pour une réponse aux détecteurs. On s'attend également à un plus haut taux de cibles localisées qu'avec l'approche centralisée du traitement au serveur car l'engouement sera situé aux têtes de grappe plutôt qu'au seul point de traitement du serveur.

4.3 Détails d'implémentations

Le logiciel de simulation utilisé est Qualnet 3.9 de Scalar Networks. Ce logiciel a été développé spécifiquement pour les besoins des développeurs de protocoles de réseautique, tant filaires que sans fil, mais avec un focus sur la simulation de protocoles pour les réseaux MANETS (Mobile Ad hoc Networks), Wimax et satellites. Le logiciel offre non seulement une librairie de protocoles modélisés pour la simulation mais de plus offre des outils de visualisation permettant d'imager les statistiques générées lors des simulations. Qualnet se compose de strates correspondant aux couches physique, liaison, réseau (IP), transport et application. Pour chaque couche, plusieurs modèles de protocoles sont proposés. Le développeur les combine à son choix dans le fichier de configuration. RCCT, le protocole proposé dans ce mémoire, est implémenté à la couche 3, réseau, de même qu'AODV, le protocole concurrent. Les couches physique et liaison sont composées des modèles pour le 802.11. Les couches application et transport ne sont pas requises pour nos simulations.

4.3.1 Configuration

Le fichier de configuration est un élément critique pour les simulations. Il permet de spécifier tous les paramètres d'opération des différents modèles. Notamment, on y spécifie le nombre de nœuds, leur vitesse, le territoire disponible et leur disposition sur ce territoire. Dans notre cas, certains paramètres ont un impact déterminant sur les résultats générés. Entre autres :

- Débit de transmission (1 - 11Mbps)
- La distance entre les cibles (200 - 400m)
- L'utilisation de RTS/CTS (802.11)
- Le nombre de retransmissions en attente d'un accusé de réception (802.11)

En effet, nous avons noté que pour des protocoles où le trafic est important comme dans ce cas-ci, le taux de réception était plus élevé pour une vitesse de transmission plus faible, 1Mbps. En effet, ceci donne plus de temps pour chaque donnée transmise et diminue le risque de corruption. La distance entre les nœuds de détection a aussi un impact important. Une distance de 250 mètres nous a semblé optimale. Dans les configurations utilisées, nous avons choisi de répartir les nœuds de détection sur une grille plutôt que de façon uniforme pour faciliter l'interprétation des résultats.

4.3.2 Méthode et variables

Le protocole implémenté se divise en deux parties : les grappes et le routage proactif. Chaque partie requière un temps d'initialisation pour les mettre en place et en état de fonctionner. Par la suite, leur fonctionnement est régi par un minimum de paquets de contrôle qui s'assurent de garder les données de chaque nœud concernant la topologie à jour. Une fois le système initialisé et en attente, les cibles sont activées et leur détection peut débuter. L'implémentation des parties critiques sera présentée.

4.3.3 Initialisation - Formation des grappes

La phase *floodmax/floodmin* permettant la sélection de la tête de grappe consiste à envoyer son meilleur candidat, basé sur son Id, par diffusion multiple dans une série d'envois dont le nombre est déterminé par le futur 'rayon' de la grappe. Cet algorithme, prévu pour les réseaux filaires, démontrait beaucoup de variabilité quant aux têtes élues dans le réseau sans fil. En effet, des nœuds voisins pouvaient avoir tous élus

une tête différente et aucun celle qui semblait le choix optimal. Le médium sans fil peut parfois permettre la communication entre deux nœuds, qui, bien qu'éloignés, parviendront occasionnellement à échanger de l'information. Ceci faussait les résultats. Pour enrayer ce comportement, chaque nœud envoie préalablement une série de messages sondes permettant d'établir une liste de voisins fiables. Lorsque les phases *floodmax/floodmin* débutent avec l'envoi de paquets MAXMIN, seuls les émetteurs figurant sur cette liste sont retenus comme candidats potentiels. Ceci régularise les résultats. Une autre modification est apportée. L'utilisation de l'adresse IP comme identificateur affecte les résultats. En effet, les nœuds étant disposés dans un ordre croissant par le simulateur Qualnet et le critère d'élection étant l'identificateur maximal reçu, cela menait à l'élection de têtes situées trop près les unes des autres. En utilisant des identificateurs pour les nœuds basés sur des numéros générés aléatoirement, tous les nœuds peuvent accéder potentiellement au statut de tête de grappe. Une fois, son choix arrêté sur une tête de grappe, le nœud détermine s'il est limitrophe, c'est-à-dire si ses voisins ont choisi une tête de grappe différente. Chaque nœud, sitôt son choix arrêté, envoie un paquet LIMIT par diffusion multiple avisant de son choix de tête de grappe. Il compile ensuite les messages reçus des voisins. Si l'un d'eux a choisi une autre tête, le nœud se déclare limitrophe et procède à aviser sa tête de grappe de son adhésion à la grappe. Il le fait en envoyant un message HEAD par diffusion unique au nœud qui à l'étape *floodmax/floodmin* lui avait envoyé la candidature de sa tête de grappe. Ce nœud, à son tour, renvoie le message vers la tête de grappe en y ajoutant son identificateur. Ce processus n'est pas optimal et une tête de grappe peut recevoir plusieurs fois l'identificateur d'un même nœud et discrimine les copies. Une fois la tête de grappe avisée de ses membres, le protocole réactif peut débuter son initialisation.

4.3.4 Initialisation - Routage proactif avec OSPF-wireless

Le protocole RCCT est inspiré en grande partie du protocole OSPF-wireless. Il requière donc l'envoi de messages HELLO pour déterminer les voisins et les relais multipoints, et de messages LSF contenant les états des liens pour déterminer la table de routage. En plus de ces deux messages, RCCT en ajoute un autre : ETX. Le paquet ETX est envoyé par diffusion multiple à intervalles réguliers. Le voisin le recevant met à jour ses statistiques de réception de paquets ETX pour ce nœud. Ainsi, six paquets

reçus en dix secondes génèrent une statistique de 60% qui décrit la qualité du lien unissant ces deux voisins. Ainsi, chaque nœud envoie des paquets HELLO, LSF et ETX et sur réception des paquets voisins, inscrit leur contenu dans des structures de données. En effet, les nœuds possèdent une structure de données pour chaque voisin. Suite à la réception d'un message HELLO d'un nouveau voisin, une nouvelle entrée est créée qui conservera toutes les données utiles correspondant à ce voisin. Notamment, on y inscrira une liste des voisins de ce voisin, ses métriques ayant trait à la qualité de lien ainsi que son état et le dernier moment où l'on a eu de ses nouvelles. Chaque paquet HELLO vient mettre à jour ces données. Un changement important dans la table de voisins engendre la génération d'un nouvel LSA décrivant l'état des liens, mais aussi le recalcul de la table de routage. De plus, les changements occasionnent aussi l'exécution de l'algorithme d'élection de relais multipoints. Une configuration différente des voisins connus indique qu'il n'est pas nécessaire de recourir aux mêmes nœuds pour retransmettre ses messages d'un saut de plus. L'algorithme déterminera les nœuds les plus aptes à retransmettre les paquets LSF.

4.3.5 Choix des relais multipoints

Les relais multipoints permettent de joindre tous les voisins à deux sauts. Leur élection est déterminée par un algorithme qui d'abord fait une liste de tous les nœuds à deux sauts de distance et leur associe tous les nœuds à un saut permettant de les joindre. Ensuite, il détermine quels voisins à un saut sont indispensables pour atteindre des voisins à deux sauts de distance et les choisit comme relais. Suite à l'ajout de ces relais, on détermine quels voisins à deux sauts sont toujours sans relais, s'en suit un processus itératif où l'on ajoute le voisin à un saut couvrant le plus de voisins à deux sauts et ce, de façon itérative, jusqu'à ce que tous les voisins éloignés par deux sauts soient couverts. Par les messages HELLO, on informe les nouveaux élus de leur statut de relais.

4.3.6 Table de LSA et algorithme de routage

Le LSA est une structure décrivant les liens entre un nœud et ses voisins directs à l'aide de la métrique ETX mais aussi qui décrit certaines métriques du nœud en soi telles sa capacité de calcul et de trafic. La structure est régénérée lorsqu'un changement important s'opère chez ses voisins tels qu'un nouveau venu ou un voi-

sin avec qui la qualité des liens s'améliore ou se détériore. Au départ, la structure LSA était régénérée incessamment car le moindre changement dans les métriques l'oblige. Ainsi, on a instauré des paliers. Tant que la qualité d'un lien se trouve dans un intervalle donné, la structure est laissée intacte. Seul le passage à un autre palier régénère le LSA. Suite à une modification du LSA, on régénère la table de routage également. Le routage se fait en parcourant tous les LSA obtenus des différents nœuds et en inspectant les liens qui y sont listés à l'aide de l'algorithme de Dijkstra. L'implémentation de Dijkstra s'est trouvée facilitée par les librairies BOOST. Ils disposent d'une implémentation qui, en recevant des arêtes et un poids pour chaque arête, en forme un arbre de recouvrement minimum. Les résultats sont emmagasinés sous la forme de paires nœud-parent. Ainsi, pour atteindre le nœud X, on doit passer par le parent Y. Si ce parent n'est pas le nœud émetteur, on suit la chaîne successivement jusqu'à parvenir à l'émetteur. Cela détermine le chemin optimal en fonction des métriques proposées.

4.3.7 Impact des intervalles

Le protocole OSPF est régi par de multiples constantes qui en influencent le comportement. Les principales constantes sont les intervalles d'émission pour les paquets HELLO, LSF et ETX. Plus ces intervalles sont rapprochés, plus le trafic de contrôle augmente et la probabilité de collisions pour le trafic utile devient importante. Les constantes concernant la gigue influencent beaucoup les résultats également. La gigue est un chiffre aléatoire qui détermine quand, dans un intervalle de temps donné, sera envoyé un message. Plus la gigue est élevé, plus on augmente l'étalement des envois dans le système et réduit la probabilité de collisions. Les temps de survie ont également une grande influence. Par exemple, le temps alloué entre deux manifestations de la présence d'un voisin avant que celui-ci ne perde son statut de voisin affecte non seulement la fréquence à laquelle le routage est recalculé mais influence comment le trafic est routé et avec quel degré de succès. En effet, ne pas enlever les voisins dont on ne reçoit plus de messages HELLO peut mener au routage du trafic utile par ce nœud même s'il n'est plus disponible. Par contre, si l'on enlève les voisins trop rapidement, on génère une surcharge de travail pour le nœud qui recalculera la table de routage régulièrement pour chaque micro variation du système.

4.3.8 Cibles - Comportement simulé

Les cibles étaient d'abord considérées des nœuds au même titre que les nœuds de détection. Toutefois, elles utilisaient également le 802.11 pour transmettre leur Id ce qui ajoutait une quantité de trafic importante sur le médium. En situation réelle, les circuits intégrés utilisés dans les RFID peuvent transmettre à 2.4 GHz mais ce n'est pas le cas type. Des cibles virtuelles ont donc été instaurées. Ces cibles de distribution uniforme étaient générées séparément. À chaque seconde, leur position est mise à jour. Étant donné que ces cibles sont virtuelles, on devait deviser un mécanisme pour déterminer quels nœuds les détectaient. Chaque nœud itère donc à travers toutes les cibles et, si celles-ci se trouvent dans un rayon de 280 mètres du nœud, il les détecte. Cette méthode ne reflète pas la variabilité du medium RFID mais cette étape de communication n'est pas sous étude ici. Une fois déterminé les cibles sous sa charge, le nœud peut donc procéder à leur localisation en partageant l'information qu'il possède sur la cible avec ses voisins.

4.3.9 Détection - Comportement

Les données de trois nœuds doivent être partagées pour permettre la localisation. Chaque cible est représentée par une structure contenant non seulement l'identificateur de la cible mais aussi une estampille temporelle reflétant le moment de détection. La localisation se fait en combinant les délais par rapport aux estampilles temporelles pour trois nœuds différents. Notre implémentation, qui simule les cibles, simule également leur traitement. Ainsi, lorsqu'on accumule trois messages de cibles, provenant de trois nœuds différents, et partageant la même estampille temporelle, on assigne cette cible-temps sur une liste de cibles prêtes pour le traitement. Cette liste est réduite par une constante, par exemple mille cibles, à chaque seconde, pour simuler que la localisation de ces cibles a été effectuée. Le protocole RCCT dicte que les nœuds de détection envoient leur cibles détectées à la tête de grappe. Celle-ci se charge du traitement lorsqu'elle obtient la même cible trois fois. Les cibles détectées sont par la suite rapportées à un serveur dans un paquet TARGET INFO. Le nœud serveur est choisi au hasard. Afin de limiter le trafic envoyé au serveur, on envoie plusieurs cibles simultanément dans chaque message. En effet, l'envoi des cibles à l'unité ne serait pas recommandable car toutes les entêtes relatives aux différentes couches OSI seraient requises à chaque fois. Notons aussi que dans ses envois à la tête de grappe,

le nœud de détection envoie lui aussi plusieurs cibles à la fois par paquet TARGET. Pour router ses paquets vers la tête de grappe, le nœud de détection utilise la table de routage. De même, pour transmettre les résultats au serveur, la table de routage est utilisée. Ainsi, on envoie les résultats au serveur en utilisant le chemin le plus court en fonction des métriques devisées. Le serveur peut agir sur l'information relative aux cibles. Dans notre cas, il génère des statistiques et sert de base de comparaison avec le protocole AODV.

4.3.10 Option - Confirmation

Le but de rapprocher le traitement du lieu actuel des cibles est double. Il consiste d'une part à diminuer le trafic jusqu'au serveur mais aussi il permet d'agir rapidement sur la cible au besoin. Pour ce faire, il est possible d'envoyer une confirmation de détection à l'un des nœuds ayant détectée la cible avec un paquet TARGET CONFIRM, lui donnant ainsi l'aval pour prendre une action déterminée par rapport à la cible. Les cibles sont détectées par de multiples nœuds de détection. Pour éviter que plusieurs nœuds dupliquent une action, on n'émet la confirmation de détection qu'au nœud détecteur ayant signalé la paire cible-temps le premier. Ainsi, lorsqu'un nœud envoie les cibles détectées à la tête de grappe et que celle-ci les enregistre dans ses listes, elle l'associe à l'identificateur du nœud de détection l'ayant rapportée le premier. Ainsi, à chaque seconde, lorsque la tête de grappe s'apprête à informer le serveur des cibles détectées, si cette option est activée, elle parcourt toutes les cibles à rapporter et les trie par identificateur de détecteur. Une fois triées, elle envoie toutes les cibles se rapportant à un même détecteur dans un même envoi. Encore une fois, la table de routage est utilisée pour déterminer à chaque étape le prochain saut pour le renvoi du paquet jusqu'au détecteur. Une fois le paquet reçu, le détecteur peut ainsi choisir de poser une action.

4.3.11 Option - Election de chargé

Lorsque la tête de grappe reçoit un nombre élevé de cibles pour traitement, elle peut devenir surchargée et éventuellement ne plus suffire à la demande. Pour éviter les engorgements, la tête se choisit un ou plusieurs chargé de traitement. Afin d'en informer ses membres, elle envoie périodiquement un message LIFE. Ce message inclut l'identificateur de la tête, l'identificateur d'une tête de relève si la tête actuelle vient

à tomber, et une liste de chargés, s'il y a lieu. Les chargés sont choisis par la tête de grappe en inspectant sa liste de membres et en extrayant pour chacun d'eux leur métrique de traitement. Elle choisira les plus avantageux, jusqu'à ce qu'ils puissent conjointement répondre à la demande de traitement. Un nœud chargé ne transmet plus ses cibles à la tête de grappe, il les envoie par diffusion générale. Les voisins les recevant lui enverront leurs cibles communes plutôt qu'à la tête. Afin de leur permettre de trier leurs messages avant l'envoi à la tête de grappe, le nœud chargé envoie ses cibles avant les autres nœuds. En instaurant un léger retard avant l' envoi de cibles à la tête de grappe, et ce afin de permettre au chargé de leur communiquer ses cibles, les nœuds peuvent réduire la charge de cibles envoyées à la tête de grappe.

4.3.12 Option - Nœud limitrophe

Les nœuds de détection situés à la limite entre deux grappes peuvent augmenter le nombre de cibles détectées en partageant l'information sur les cibles détectées. Ainsi, un nœud limitrophe enverra ses cibles par diffusion multiple plutôt qu'unique. Le nœud qui relaie normalement ses données vers la tête le fera sur réception du message mais, de plus, un nœud limitrophe appartenant à une autre grappe, les relaiera lui aussi à sa propre tête de grappe. Ceci augmente les chances de détection de cibles situées entre deux grappes.

4.4 Plan d'expérimentation

Nous souhaitons d'abord démontrer que le protocole RCCT est plus performant qu'AODV à l'état pur. Ainsi, le test initial consiste en l'envoi des cibles directement au serveur avec AODV. Par la suite, nous étudierons les différences engendrées par l'ajout des grappes avec un algorithme tel qu'AODV. Puis nous comparerons ces solutions avec RCCT où une variante d'OSPF se chargera des tâches de routage. Nous observerons également les améliorations apportées par l'utilisation des nœuds limites ou des nœuds chargés. Nous vérifierons aussi l'efficacité de la confirmation au détecteur avec la méthode AODV ainsi qu'avec la méthode RCCT avec confirmation. Le facteur principal qui sera varié sera le nombre de cibles. Les valeurs mesurées seront le nombre de cibles-temps localisées, le délai moyen requis, le nombre de messages moyen requis ainsi que la proportion de messages de contrôle requis.

4.5 Résultats

Cette étude évalue différentes variantes du protocole AODV et RCCT pour la localisation maximale de cibles. La première partie se concentre donc sur les performances des protocoles pour un nombre variables de cibles. Voici les paramètres communs des différentes simulations pour ce premier test :

- Vitesse de transmission des données : 1 Mbps
- Distance entre les nœuds : 250 mètres
- Utilisation de CTS/RTS pour paquets par diffusion unique : Oui
- Vitesse des cibles : 0 m/s
- Envoi des cibles : 15 cibles maximum par paquet
- Temps de simulation : 20 minutes

Le premier test détermine le nombre de cibles qui parviennent au serveur et est illustré à la Figure 4.1. On y voit le nombre de cibles envoyées par les protocoles suivants :

- AODV directement (individuellement) ;
- AODV directement (cibles par groupe de 15) ;
- AODV avec usage de têtes de grappe ;
- RCCT avec usage de têtes de grappe.

La première constatation est qu'avec AODV, lorsque chaque cible est envoyée directement au serveur individuellement, on obtient des performances insatisfaisantes déjà pour une quantité de 100 cibles dans le réseau. On parvient à améliorer grandement cette performance lorsque les nœuds envoient les cibles détectées en les regroupant ensemble par paquets de 15 cibles détectées. Ainsi, un nœud réduit par un facteur de quinze le nombre de ses envois de données utiles. On remarque aussitôt que les performances s'en trouvent affectées. On passe pour une configuration de 100 cibles de 24000 cibles-temps à 53000 localisations. Cette proportion devient encore plus importante pour 500 cibles alors que les détections passent de 800 cibles-temps, pour un réseau saturé, à 160000 localisations. Cette amélioration se doit à moins de trafic, ce qui engendre bien sûr moins de collisions mais aussi permet d'assurer le maintien des routes et évite le trafic requis pour la redécouverte de nouvelles routes pour acheminer les paquets vers le serveur.

Dans un second temps, les tests sont effectués pour observer le comportement d'AODV lorsque des grappes sont ajoutées au protocole pour rapprocher le traitement

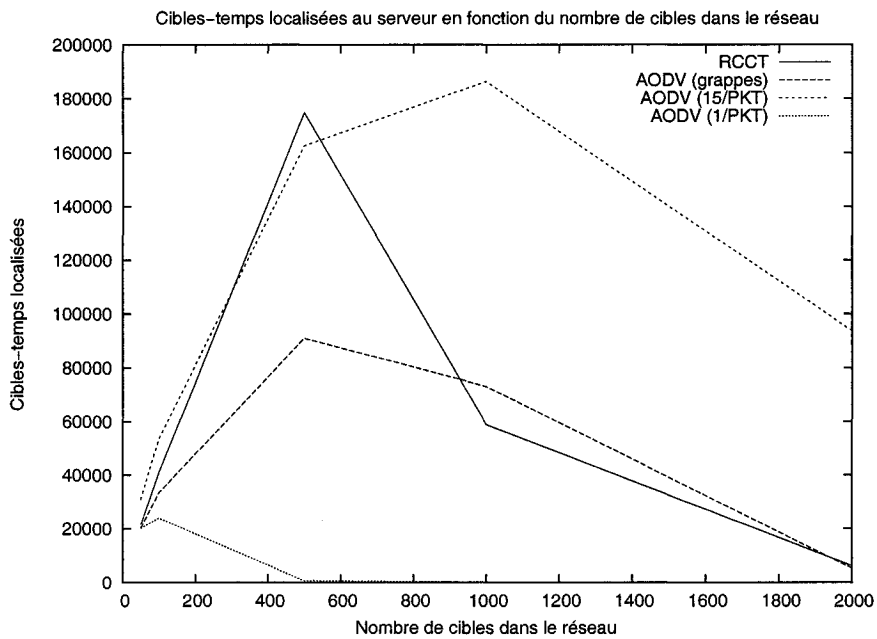

FIGURE 4.1 Cibles-temps localisées au serveur

des détecteurs plutôt que du serveur. Cette modification doit réduire le trafic car les têtes de grappe sont plus proches des nœuds de détection que ne l'est le serveur. Ainsi, un allègement du trafic se produit car il en coûte moins d'envoyer une donnée de cible localisée au serveur que de lui envoyer les données brutes dues à toutes les observations de la cible par plusieurs nœuds de détection. Toutefois, on observe que cette modification ne produit pas d'améliorations, bien au contraire. AODV avec traitement au serveur produit de meilleurs résultats. À titre d'exemple, pour 500 nœuds, il localise 160000 cibles-temps alors qu'AODV avec grappes n'en localise que 90000. On peut toutefois expliquer ce résultat du fait que le réseau est de petite taille, avec 25 nœuds de détection, et que le serveur se trouve à une distance raisonnable des nœuds de détection. Aussi, en raison de la proximité du serveur, il résulte plus lourd d'envoyer toutes les données à une tête de grappe pour que celle-ci renvoie à nouveau les données traitées vers le serveur. Cela génère un double trafic. On peut toutefois supposer raisonnablement que, pour un réseau de grande taille où une tête de grappe se trouve à proximité des nœuds de détection, le trafic se trouverait diminué plutôt qu'amplifié avec l'usage des têtes de grappe.

Dans un troisième temps, RCCT, le protocole proposé, est utilisé pour acheminer les cibles-temps localisées au serveur. Encore une fois, un algorithme de grappe est utilisé pour l'élection de têtes de grappe. On observe qu'AODV (15/PKT) démontre cette fois encore de meilleures performances hormis le cas des 500 cibles où les deux protocoles semblent performer de façon équivalente, avec une légère avance pour RCCT. Il est à noter toutefois que RCCT performe mieux qu'AODV avec grappes. RCCT, en effet, met à la disposition de tous les nœuds du réseau, une table de routage listant tous les nœuds du réseau et l'état des liens de ces nœuds. Une fois la table formée, un certain trafic de contrôle est encore requis pour mettre à jour l'état des liens avec l'envoi de paquets Hello, LSF et ETX. AODV, toutefois, n'établit les routes qu'en cas de besoin, à l'aide de paquets RREQ et RREP. Avec AODV, un nœud s'assure que le voisin qui forme le prochain saut pour une de ses routes est toujours actif. Il attend des paquets ACK de ses voisins ou encore des paquets HELLO. Si ce voisin tarde à envoyer ces messages, le nœud peut choisir de former une nouvelle route et de retirer ce voisin de celles-ci. Il déclenche alors un processus de recherche de route coûteux en termes de paquets. On note que pour les cas de 1000 et 2000 cibles, AODV avec grappes et RCCT performent similairement avec une légère avance d'AODV. Cette avance peut être due au fait que RCCT envoie ses paquets périodiques de mises à jour des liens à intervalles trop grands ce qui ralentit la réactivité de RCCT en cas de changements de qualité des liens.

4.5.1 Nombre moyen de messages de contrôle requis par localisation au serveur

Le nombre de messages de contrôle requis en moyenne pour l'acheminement au serveur est étudié. La Figure 4.2 inclut les messages de contrôle totaux pour offrir un meilleur reflet des tendances. On peut y voir que lorsqu'AODV envoie une cible par paquet, l'importance du trafic sature les communications et AODV tente par tous les moyens de générer de nouvelles routes. À titre d'exemple, le nombre moyen de paquets de contrôle requis pour 500 cibles est de 1650 paquets de contrôle par cible localisée. On remarque tout de suite, qu'en envoyant les cibles localisées en tranches de quinze à la fois, le réseau supporte beaucoup mieux le trafic et requière un échange de paquets de contrôle moins important car moins de routes nouvelles sont requises, les routes actuelles demeurant valides plus longtemps. Ainsi, pour 500 cibles

dans le réseau, le nombre moyen de paquets de contrôle requis pour une localisation tombe à 1,5. Lors de l'utilisation de grappes, AODV utilise en moyenne 2 paquets de contrôle par localisation acheminée au serveur. On avait noté plus haut que la baisse de performance d'AODV avec grappes pouvait être due à une augmentation de la recherche de nouvelles routes.

Cette augmentation du rapport de paquets de contrôle par cible localisée semble pointer effectivement dans cette direction. Quant à RCCT, le nombre de paquets de contrôle tend à être constant à 14000 par simulation. En effet, l'envoi de paquets de contrôle n'est pas motivé par l'état du réseau mais plutôt par la mise à jour périodique de l'état des liens. Cette périodicité garantit un nombre maximum de paquets de contrôle dans le réseau. Ceci limite un peu la flexibilité de la solution car dans un réseau en changement constant, la mise à jour des liens périodique omettrait certains changements se produisant entre deux mises à jour. Toutefois, en reflétant l'état global du réseau, elle offre des solutions alternatives aux routes utilisées. De plus, cette méthode évite l'explosion de paquets de contrôle dans le réseau pour la formation de nouvelles routes.

4.5.2 Nombre moyen de messages de données requis par localisation au serveur

Le nombre de messages requis en moyenne pour la localisation au serveur est étudié et illustré à la Figure 4.3. On observe que le nombre de paquets est, sans surprise, très élevé lorsqu'AODV envoie chaque cible détectée individuellement. Cette différence n'est pas quinze fois plus élevée car certains nœuds n'atteignent pas les quinze cibles détectées à chaque seconde. Il faut noter aussi que chaque paquet de données ne résulte pas en une localisation car le paquet ne sera pas nécessairement reçu. À titre d'exemple, pour 500 cibles, le nombre de paquets de données envoyé est trois fois plus élevé entre AODV (15/PKT) et AODV (1/PKT). En moyenne, AODV (15/PKT) utilise 2 paquets de données pour effectuer une localisation, AODV (grappes) en utilise 3,5 et RCCT en utilise 1,5 pour un réseau de 500 cibles. Il se peut que RCCT utilise moins de paquets de données car les cibles localisées se produisent simultanément et sont envoyées dans un même paquet. En combinant plusieurs cibles par envoi, cela permet de réduire le nombre de paquets de données de façon importante.

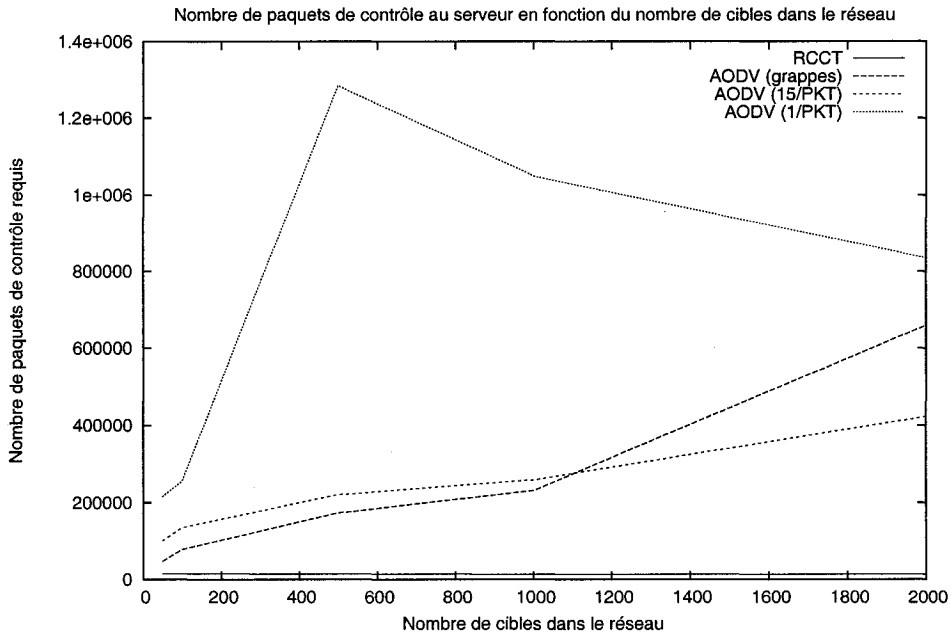

FIGURE 4.2 Nombre moyen de messages de contrôle requis par localisation au serveur

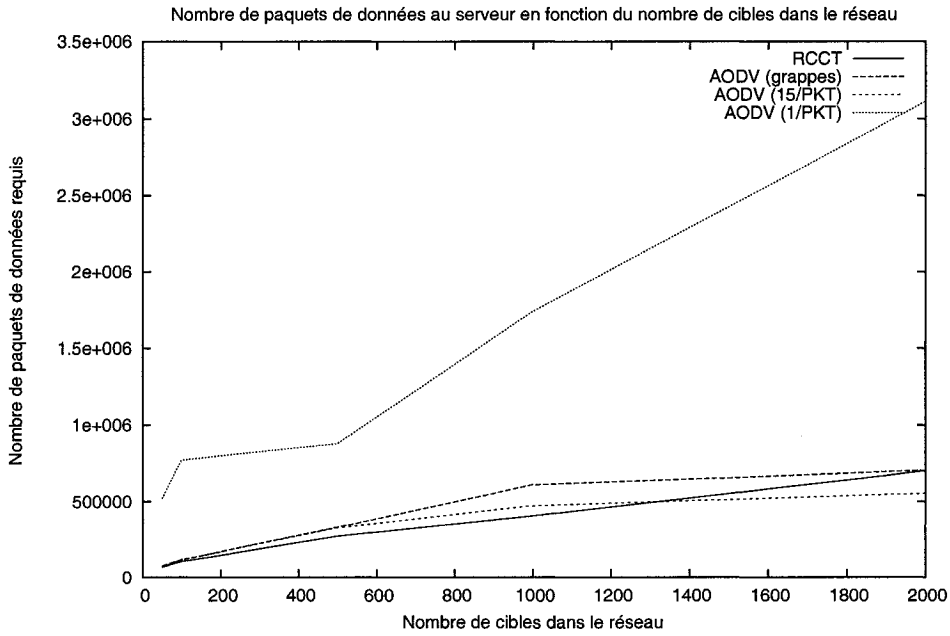

FIGURE 4.3 Nombre moyen de messages de données requis par localisation au serveur

4.5.3 Délai moyen requis par localisation au serveur

Le délai moyen requis pour localiser une cible pour le serveur est étudié et illustré à la Figure 4.4. Ce délai n'est représenté que pour AODV (grappes) et RCCT. On peut y apercevoir que le délai moyen est plus court avec AODV (grappes) que RCCT. Pour 500 cibles, le délai moyen est de 1,5 seconde chez AODV et 3,7 secondes pour RCCT. La différence entre AODV (grappes) et RCCT se doit à quelques cibles qui parviennent au serveur après de multiples retransmissions. Bien que ces cibles soient peu nombreuses, le retard important dont elles font preuve affecte la moyenne des délais. Il y a plus de cibles retardataires avec RCCT, car la route utilisée est faible alors qu'AODV ne parvient pas à établir de route pour ce trajet et rejette le paquet plus tôt. À titre de référence, pour AODV (15/PKT) le délai se situe pour 500 cibles à 0,6 seconde. On peut facilement s'expliquer ce résultat car les cibles sont traitées sur place et non pas traitées par une tête de grappe avant d'être réacheminées vers le serveur. Cela a pour effet de diminuer les délais.

4.5.4 Cibles localisées au détecteur

Le deuxième test utilise la même configuration et les mêmes protocoles et s'intéresse au nombre de cibles localisées au détecteur et est illustré à la Figure 4.5. Les trois protocoles étudiés plus haut sont évalués pour un nombre variable de cibles. Cette simulation est pertinente afin de refléter le cas où les nœuds de détection veulent agir suite à la localisation d'une cible, en déclenchant, par exemple, une alarme. Dans ce cas-ci, l'utilisation de grappes devrait être profitable car cela rapproche le traitement des données des nœuds de détection. En ce cas, les données ont moins de distance (sauts) à parcourir pour parvenir à la tête de grappe ainsi que moins pour en revenir lorsque les données sont traitées, par rapport à un serveur. On y remarque que RCCT est le protocole le plus avantageux en termes de cibles-temps localisées. Cet avantage est surtout marqué pour une quantité d'environ 500 cibles. On peut s'expliquer cet avantage par rapport à AODV (15/PKT) par la proximité entre les cibles et les têtes de grappe. Par rapport à AODV (grappes), l'explication se doit au fait qu'AODV doit reformer ses routes chaque fois qu'un nœud détermine qu'elles sont trop instables et que ce processus est coûteux en paquets et engendre à son tour de nouvelles collisions suite à ce trafic de contrôle.

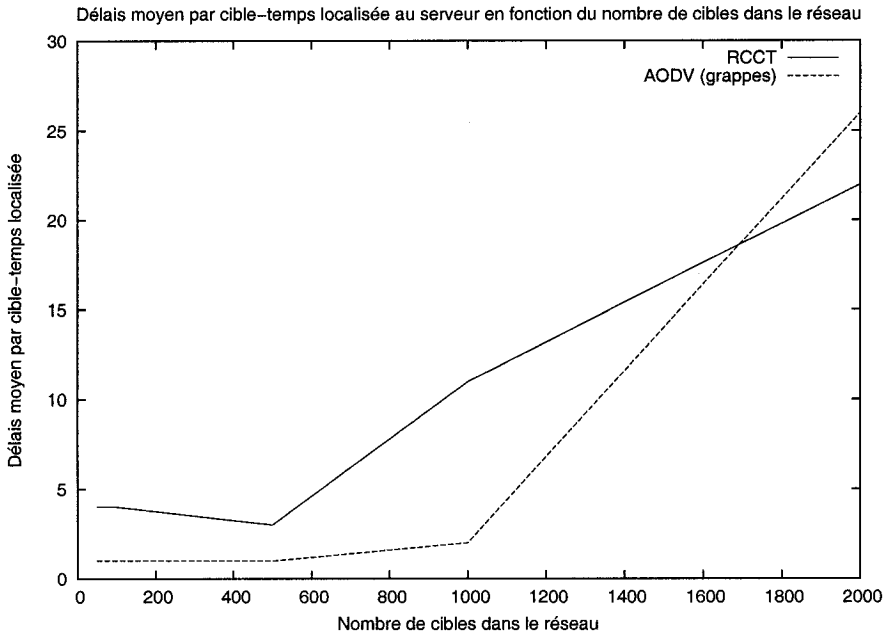

FIGURE 4.4 Délai moyen requis par localisation au serveur

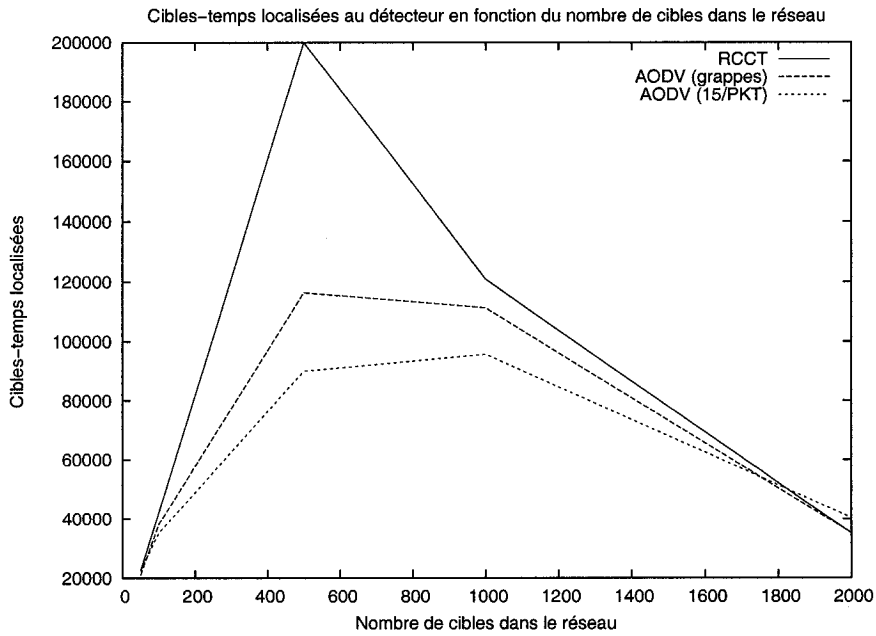

FIGURE 4.5 Cibles localisées au détecteur

4.5.5 Nombre moyen de messages de contrôle requis par localisation au détecteur

Le nombre de messages de contrôle requis en moyenne pour la localisation au détecteur est étudié et illustré à la Figure 4.6. Encore une fois, on observe que la quantité est constante pour RCCT et qu'elle varie de façon importante pour AODV suite aux ajustements requis pour la découverte de routes sur demande. Pour 500 cibles, on observe que RCCT requière 0,06 messages de contrôle, AODV (grappes) en requière 1,5 et AODV (15/PKT) 2,5. Pour la quantité de cibles localisées obtenue, la quantité constante de paquets de contrôle de RCCT maintient à un minimum ce rapport.

4.5.6 Nombre moyen de messages de données requis par localisation au détecteur

Le nombre de messages de données requis en moyenne pour l'acheminement au détecteur est étudié et illustré à la Figure 4.7. Par rapport à l'acheminement au serveur, on s'attend à une diminution du nombre de paquets de données car la distance à parcourir est réduite. On observe, pour 500 paquets, que la moyenne pour AODV (15/PKT) est de 3,6, de 2,1 pour AODV (grappes) et de 1,2 pour RCCT. Les valeurs totales sont similaires. C'est toutefois lorsque ramenées sur une base par cible localisée que la différence se fait plus sentir. Cela indique que pour AODV plusieurs cibles ne se rendent pas à destination bien qu'elles soient envoyées.

4.5.7 Délai moyen requis par localisation au détecteur

Le délai moyen requis pour localiser une cible pour le détecteur est étudié. On observe que pour 500 cibles, on obtient un délai moyen de 3,1 secondes pour RCCT, 1,4 pour AODV (grappes) et 1,6 pour AODV (15/PKT). Encore une fois, RCCT impose des délais plus élevés aux cibles-temps localisées. Cela se doit en partie à une minorité de cibles avec un important retard dû aux retransmissions. On note toutefois que les délais diminuent légèrement par rapport à l'envoi au serveur pour RCCT et augmente pour AODV (15/PKT) en raison du traitement initial au serveur pour AODV avant l'envoi au détecteur.

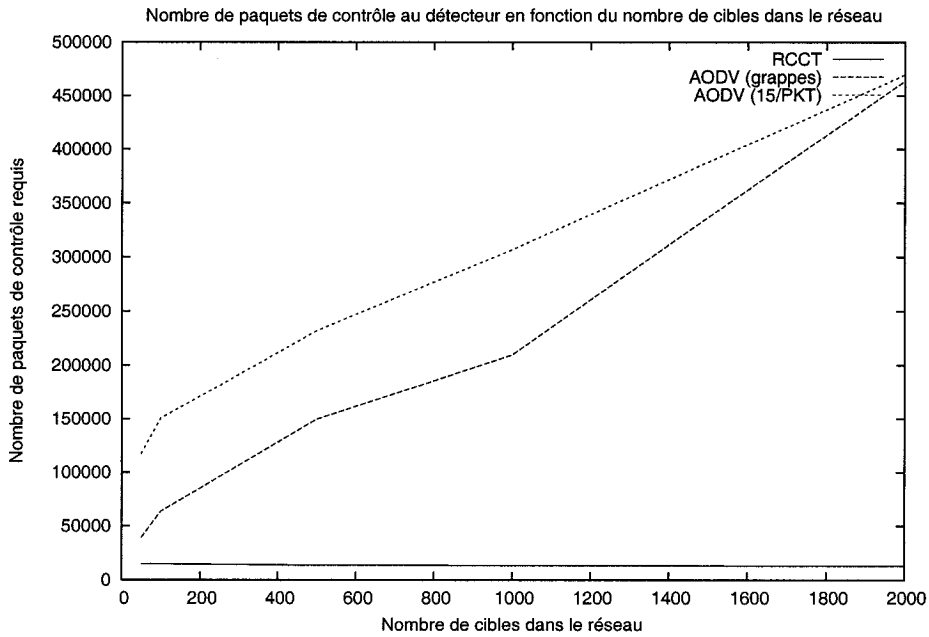

FIGURE 4.6 Nombre moyen de messages de contrôle requis par localisation au détecteur

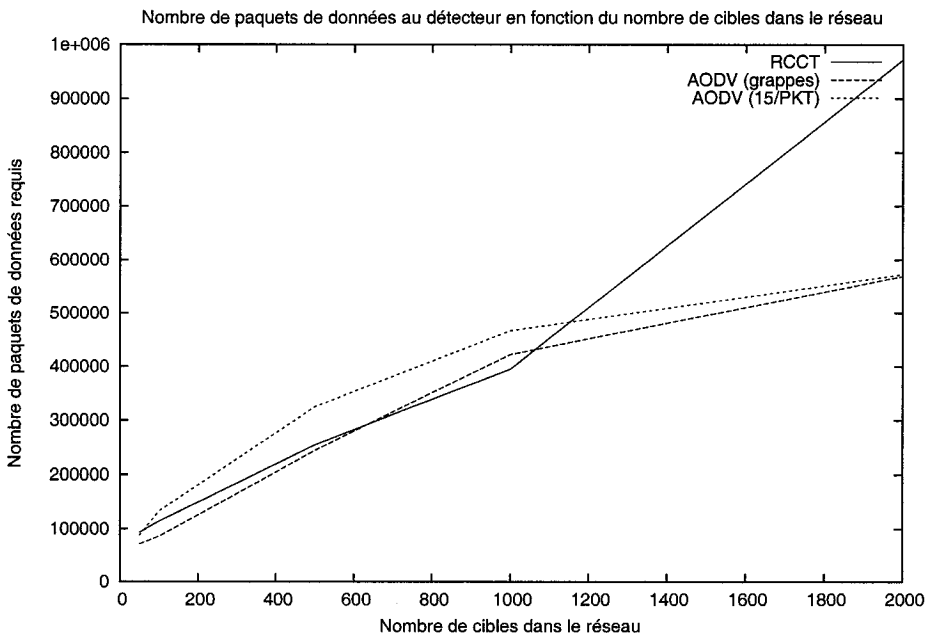

FIGURE 4.7 Nombre moyen de messages de données requis par localisation au détecteur

4.5.8 Algorithme Max-Min et impact sur localisation de cibles-temps

L'algorithme Max-Min choisi est affecté par de multiples facteurs, dont la distance entre les nœuds, leur capacité de transmission, le bruit dans l'environnement ainsi que l'attribution des identificateurs qui est la base même du mécanisme d'élection de têtes de grappe. Suite au choix de ces paramètres, on peut évaluer si les têtes de grappe choisies se révèlent performantes pour obtenir un maximum de cibles localisées. On peut varier le nombre de bonds maximum qui relient une tête de grappe à ses nœuds membres. Les tests effectués, variant le nombre de bonds maximum, démontre que pour le réseau donné de 25 nœuds, le choix le plus performant est un rayon de 2 bonds. Les résultats sont, pour 600 cibles présentes, de 153000 cibles-temps au serveur pour 2 bonds, 145000 cibles-temps au serveur pour 1 bond et 89000 cibles au serveur pour 3 bonds. On peut aisément expliquer que le choix de 3 bonds soit peu intéressant pour un réseau de 25 nœuds. En effet, cela mène à l'élection d'une seule tête de grappe. Ainsi, cette tête de grappe reprend le rôle en quelque sorte d'un serveur en desservant tous les nœuds avec le désagrément ajouté de devoir renvoyer l'ensemble des cibles-temps localisées au serveur actuel. La solution à un bond génère cinq têtes de grappe, contre deux pour la solution à deux bonds. On peut expliquer la légère avance de la solution à deux bonds par le fait que les cibles situées à la bordure de leur grappe peuvent être détectées par des nœuds appartenant à différentes grappes ce qui réduit leur chance de détection, chaque nœud rapportant ses détections uniquement à sa tête de grappe.

4.5.9 Détection et nœuds limitrophes

Les nœuds limitrophes sont étudiés et les résultats illustrés à la Figure 4.8. Les nœuds situés en bordure de deux grappes différentes peuvent partager leurs données afin de pourvoir les deux grappes de données suffisantes pour faciliter la localisation. Pour ce faire, ils n'envoient pas directement leurs données à la tête de grappe mais les diffusent. Les nœuds limitrophes de grappes adjacentes, ayant un identificateur plus élevé, peuvent ainsi envoyer les cibles communes à leur tête de grappe. Les tests effectués vérifient l'efficacité de cette méthode par la localisation de plus de cibles-temps. Toutefois les résultats ne sont pas probants. En effet, bien qu'il soit possible que plus de cibles ne soient traitées suite à ce partage de données, l'augmentation de

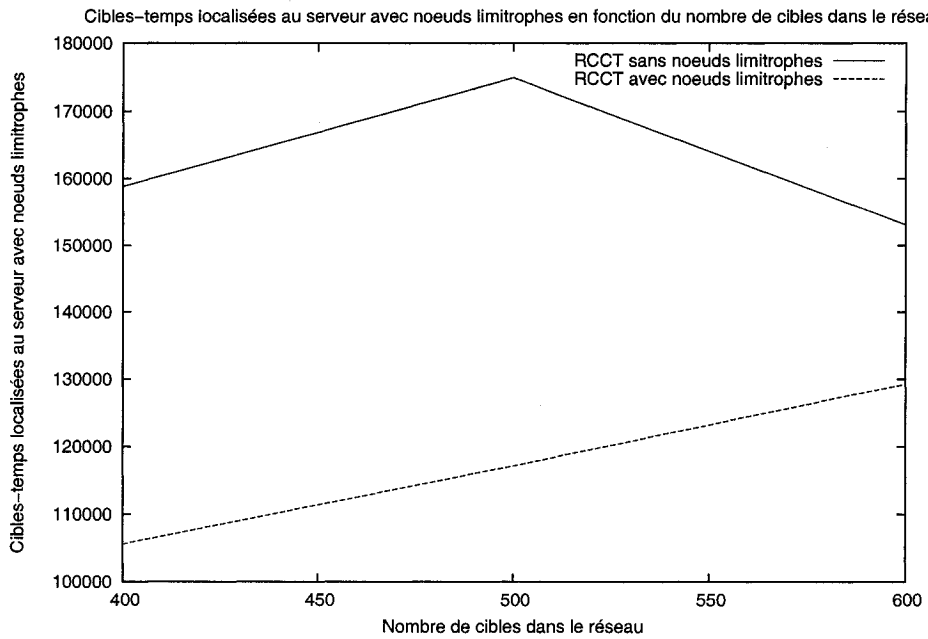

FIGURE 4.8 Nombre de cibles-temps localisées selon le nombre de cibles dans le réseau

paquets à traiter par les nœuds limitrophes ne semble pas justifier l'amélioration. En effet, le nombre de localisations effectuées avec succès, dans un réseau comprenant 500 cibles, est de 117000 lorsque les nœuds limitrophes échangent des données et de 175000 lorsqu'ils ne partagent pas d'informations. Il faut noter toutefois qu'en raison du rapprochement important des nœuds de détection, 12 nœuds de détection sur les 25 disponibles se déclarent limitrophes. Cela engendre une augmentation importante du nombre de paquets de données envoyées qui, pour 500 cibles, passe de 270000 à 636000 lorsque les nœuds limitrophes sont actifs. Il est possible qu'en raffermissant le critère de sélection pour devenir limitrophe, le nombre d'échanges résultant plus limité favorise l'augmentation de cibles-temps localisées.

4.5.10 Impact des métriques sur le routage

L'influence des métriques sur la formation de la table de routage est étudié et illustré à la Figure 4.9. En effet, lorsque tous les LSA accumulés par un nœud sont parcourus par l'algorithme de Dijkstra, le poids de chacune des arêtes est déterminé à l'aide des métriques de qualité de liens calculées suite à l'échange de paquets ETX, la

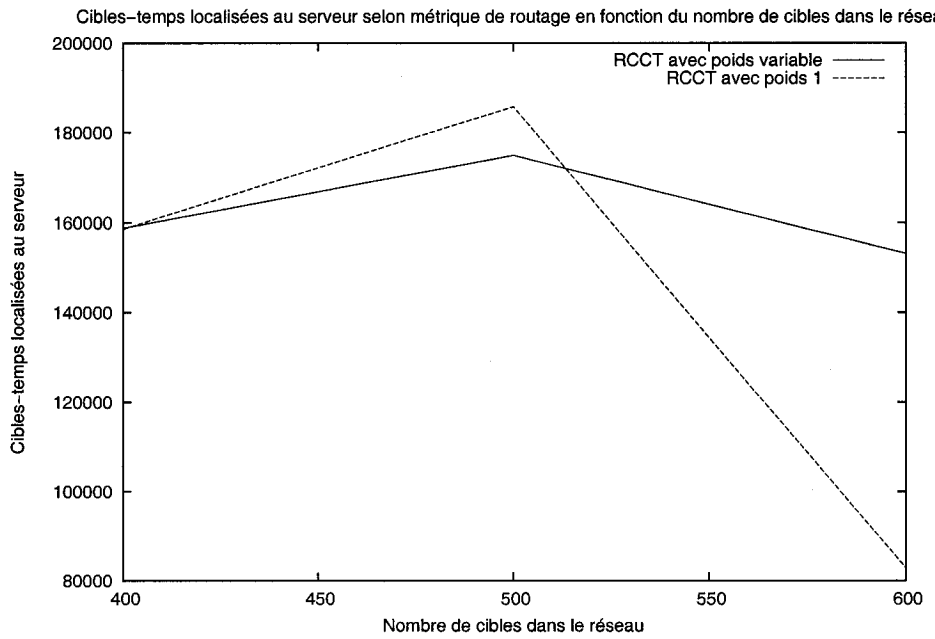

FIGURE 4.9 Nombre de cibles-temps localisées selon le nombre de cibles dans le réseau

métrique de quantité de liens n'étant pas utilisée car elle changeait peu. ETX envoie des paquets sondes à intervalles réguliers et sur la proportion de paquets sondes reçus, le nœud établit des statistiques sur la qualité des liens que le nœud partage avec ses voisins. Le premier test compare l'impact sur la formation des tables de routage en vérifiant quel mécanisme est plus efficace : un poids unique arbitraire de 100 pour chaque arête de l'arbre de recouvrement minimum, ou encore des poids situés entre 1 et 100 pour les arêtes, poids spécifié par les statistiques ETX. Suite aux simulations effectuées, on peut observer que les deux méthodes obtiennent des résultats similaires pour une quantité de cibles de 400 nœuds avec 158000 cibles-temps détectées. Pour 500 nœuds, le poids unique l'emporte légèrement avec 185000 détections contre 175000 pour le poids variable. La mince différence peut se devoir à des routes plus longues, au poids plus faible que des routes plus courtes. Par contre, pour 600 nœuds, alors que le trafic engendré devient plus important, on note que le poids unique a un impact néfaste nuisant à l'acheminement des cibles-temps. En effet, pour 600 cibles, les détections se situent autour de 153000 pour la méthode par poids variable et de 89000 pour la méthode par poids fixe. On peut en déduire que le trafic important nuit aux routes plus courtes mais ayant un taux de succès plus faible.

4.5.11 Détection et chargés de traitement

L'impact des chargés de traitement est étudié et illustré à la Figure 4.10. Lorsque le réseau devient chargé, le trafic aux têtes de grappe augmente et peut résulter en paquets rejetés. Afin d'éviter cette situation, les têtes de grappe envoient à intervalles réguliers des paquets LIFE rapportant les chargés de traitements, s'il y en a. Ceux-ci sont choisis pour leur ample capacité de traitement, qu'ils mettent à disposition des têtes de grappe pour effectuer des traitements de cibles. Le chargé de traitement, choisi par une tête de grappe, diffuse ses cibles plutôt que les acheminer directement vers la tête de grappe. Les nœuds voisins qui partagent des cibles avec les nouveaux chargés les enverront à ceux-ci plutôt qu'à leur tête de grappe, réduisant ainsi le fardeau de travail de la tête de grappe. Afin d'informer leurs voisins avant qu'ils n'aient déjà envoyés leurs cibles à la tête de grappe, un léger délai est ajouté afin que les voisins puissent recevoir les cibles à temps. Les résultats sont probants. On observe que, pour 500 cibles dans le réseau, le nombre de cibles-temps détectées avec succès passe de 175000 à 185000. C'est encore plus probant lorsque le nombre de cibles passe à 600, on observe alors une augmentation de localisations de cibles-temps de 153000 à 182000. Quant au délai moyen de localisation, il n'augmente pas malgré le léger retard instauré pour l'envoi des cibles des chargés. En raison de la diminution de l'engorgement aux têtes de grappe, les paquets sont acheminés avec moins de retransmissions, résultant en une diminution du délai moyen de 3,7 à 3,5 secondes par cible-temps localisée.

4.5.12 Question de cibles

Le taux de détection des cibles n'est pas de 100%. Quel suivi des cibles peut-être assuré ?

En effet, si chaque cible était détectée à chaque seconde, on obtiendrait 600000 plutôt que 175000 pour un réseau de 500 cibles. La disposition des cibles y est pour quelque chose. En effet, on peut observer que la majorité des cibles situées en bordure ne sont JAMAIS détectées. Cela est dû au fait que le rayon de détection stipulé à 280 mètres n'est adéquat que pour les cibles entourées de nœuds de détection. Celles en périphérie n'ont pas assez de nœuds de détection à portée pour accumuler trois lectures, ainsi elles diminuent la proportion de cibles localisées. Il faut donc ajouter plus de nœuds de détection en périphérie pour permettre la détection de toutes les cibles présentes.

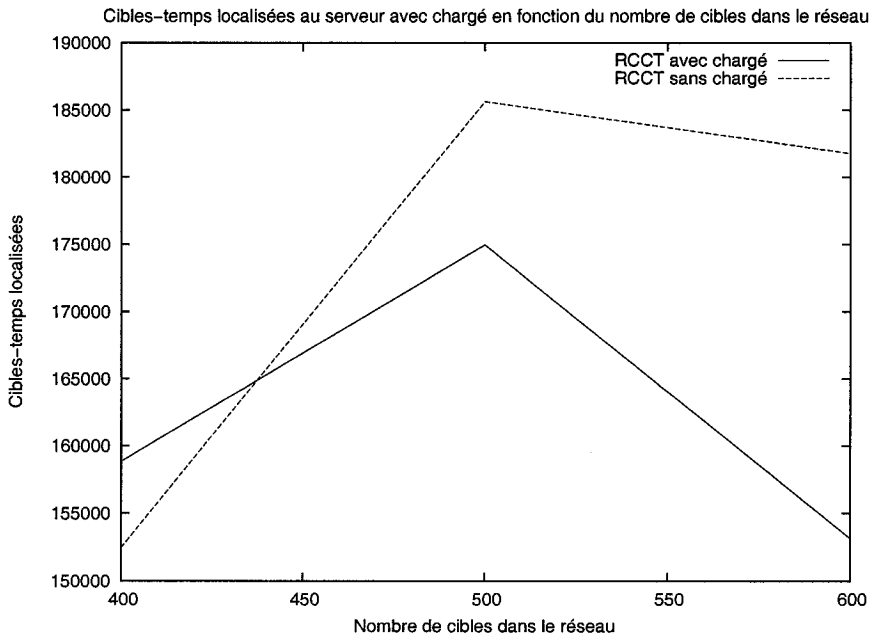

FIGURE 4.10 Nombre de cibles-temps localisées selon le nombre de cibles dans le réseau

4.6 Sommaire des résultats

En raison de l'utilisation d'une vitesse de transmission de données de seulement 1 Mbps, le nombre de paquets transmis semble atteindre des performances maximales vers les 500 cibles dans le réseau. On note que les chargés peuvent permettre de maintenir le niveau de détection lorsque le réseau devient plus chargé. Les nœuds limitrophes quant à eux génèrent un trafic important qui nuit aux détections. L'utilisation des poids pour l'algorithme de routage est surtout utile lorsque le nombre de cibles dans le réseau augmente mais ne nuit toutefois pas. Lorsque les cibles sont destinées au serveur, AODV (15/PKT) semble offrir les performances les plus intéressantes. Toutefois, pour une réponse vers les détecteurs, l'utilisation de RCCT augmente le nombre de localisations.

CHAPITRE 5

Conclusion

Dans ce mémoire, il a été question du problème de localisation de cibles. Cette question permet de considérer une application précise dans le contexte des communications sans fil et amène son lot de complexités propres. En effet, au-delà des questions de routage et de choix de têtes de grappe, la mesure réelle de succès se fait en termes de cibles-temps localisées. Ce mémoire a investigué une approche originale qui combinait la formation de grappes et le routage proactif inspiré de OSPF-wireless. Les deux composantes offrent des avantages et des inconvénients qui deviennent apparents dans les simulations présentées au chapitre précédent. Ce chapitre présente les propositions avancées dans ce mémoire et une synthèse des résultats obtenus suite aux simulations. Il suggère certaines limitations observées lors de l'analyse de la solution et propose des pistes pour la poursuite des travaux afin de permettre de meilleures performances futures.

5.1 Synthèse des travaux et originalité des contributions

Généralement, les solutions traditionnelles de traitement de cibles proposent une approche centralisée. En effet, toutes les données brutes sont acheminées intégralement au centre de traitement pour qu'une machine serveur aux capacités de traitement importantes se charge de les trier et les traiter rapidement. Parfois, toutes les machines sont identiques ou encore la machine serveur ne sert qu'à enregistrer les données préalablement traitées. Dans ce contexte, il est souhaitable que le traitement se fasse plutôt à même les nœuds du réseau. Tout en requérant plus d'énergie afin d'effectuer le traitement des cibles, on réduit le trafic généré en rapprochant les nœuds de traitement de l'endroit où se produit l'évènement, c'est-à-dire la détection de la cible. Il devient alors possible de renvoyer rapidement les résultats de la localisation au détecteur pour qu'il puisse agir, comme pour déclencher une alarme.

Afin de rapprocher le traitement des cibles, l'utilisation de grappes est un choix logique. Cela permet d'éliminer un dialogue fastidieux entre les nœuds de détection pour déterminer à qui revient le traitement de chacune des cibles détectées. En instaurant une hiérarchie, un seul nœud par grappe prend les décisions. Il se charge du traitement de cibles mais peut décider également de choisir le nombre de chargés lui étant nécessaires pour mener à bout le traitement du nombre le plus grand possible de paquets. L'algorithme choisi pour le choix des têtes de grappe est Max-Min, développé pour les réseaux filaires. La variabilité de la qualité des communications dans un réseau sans-fil cause l'élection de têtes de grappe mal réparties. Afin d'ajuster cet algorithme pour le domaine sans-fil, on précède l'échange de paquets *floodmax/floodmin* par des paquets sondes servant à déterminer une liste de voisins de qualité. Ceci assure que le choix des têtes de grappes est prévisible et constant d'une simulation à l'autre. Une fois les têtes de grappe choisies, il faut réacheminer vers un serveur, à fin d'enregistrement des données, ou vers le nœud détecteur, pour la prise d'action. Ce réacheminement peut se faire avec un protocole réactif qui bâtit les routes sur demande tel qu'AODV. Un protocole proactif a plutôt été choisi. Le choix s'est arrêté sur OSPF-wireless auquel on a ajouté des métriques de qualité de liens. Le choix d'un protocole proactif s'est fait basé sur la stabilité des nœuds de détection. En effet, ceux-ci sont immobiles. Seules les cibles peuvent se déplacer. Ainsi, les changements devraient être mineurs une fois la table de routage établie. Ainsi, après un temps d'initialisation, où les nœuds construisent leur table de voisins et choisissent les nœuds de relais multipoints qui transmettront leurs messages d'états de liens, le nombre de messages se résume aux mises à jour de l'état des liens. On voit dans les résultats présentés que le limite supérieure au nombre de paquets de mise à jour envoyés semble constituer une avantage en prévenant l'explosion des messages de découverte de routes lorsque le trafic augmente.

Les métriques de qualité de liens utilisées sont le résultat d'un échange de messages ETX. On établit des statistiques basées sur le nombre de paquets sondes reçus correctement. Cela permet de calculer les chemins pour le routage en tenant compte de la qualité des liens. On observe toutefois dans les résultats que cette métrique n'améliore les résultats que lorsque le nombre de nœuds devient plus important. Il semble qu'assigner une valeur égale à tous les liens, sans tenir compte des différences de qualité de liens, produit des résultats satisfaisant en terme de cibles-temps localisées pour un nombre moyen de cibles dans le réseau. Par contre, la performance d'un

routage avec poids de valeur unitaire décroît rapidement lorsque le nombre de cibles devient plus important.

Le choix d'un protocole proactif implique une connaissance de l'état du réseau pour chaque nœud de détection, tant pour les têtes de grappe que pour les membres de la grappe. Cela apporte de nouvelles possibilités. Avec ces connaissances, la tête de grappe, lorsque surchargée, peut choisir, parmi ses membres, les plus aptes à la seconder dans le traitement de cibles. Elle peut ainsi les nommer chargés de traitement. Les résultats indiquent que pour un nombre moyen de cibles comme pour un nombre élevé de cibles, cette capacité de choisir des voisins pour alléger la charge de traitement de la tête de grappe permet d'augmenter le nombre de localisations. Le protocole proactif, en transmettant non seulement l'état des liens par un LSA mais en y incluant aussi une métrique de charge de traitement, permet de faire des choix justifiés en termes de chargés de traitement.

Dans l'ensemble, la combinaison de la formation de grappes et du protocole proactif permet de générer des résultats intéressants dans la mesure où les résultats de la localisation sont destinés aux nœuds détecteurs. Lorsque le réseau est petit, comme dans le cas testé avec 25 nœuds de détection, et que les données serviront au serveur, il est plus intéressant d'envoyer les données brutes directement au serveur à l'aide d'un protocole comme AODV. Toutefois, il importe de regrouper les données de plusieurs cibles afin de réduire la quantité de paquets envoyés au serveur. Quand, toutefois, les données sont destinées aux détecteurs et que le volume de cibles est important, RCCT semble un choix intéressant.

5.2 Limitations des travaux

L'implémentation de l'algorithme de formation de grappes et l'implémentation du protocole OSPF-wireless respecte en tous points les documents définissant leurs comportement et caractéristiques et ceci afin de s'assurer que les performances maximales puissent être atteintes. Toutefois, la gestion des cibles pour leur localisation laisse certaines questions ouvertes. Notamment, la distribution des cibles utilisée pour nos recherches est aléatoire. Ainsi, les cibles sont généralement distribuées de façon équilibrée sur tout le territoire disponible. Le mécanisme de détection initiale des cibles n'était pas à l'étude dans ce mémoire du fait qu'il dépend d'une technologie différente (RFID). Ce processus a donc été simulé. Chaque nœud itérait à travers

toutes les cibles et déterminaient celles qui se trouvaient dans sa région de détection. Ce mécanisme simple a toutefois permis de déterminer que lorsque les nœuds de détection ne sont pas sur la bordure même du territoire disponible, les cibles s'y trouvant ne seront pas détectées par les trois nœuds de détection requis pour la localisation. Afin de faciliter l'interprétation des résultats, les cibles étaient immobiles. Cette simplification de la réalité des cibles peut teinter légèrement les résultats obtenus, sans toutefois avantager une solution plus qu'une autre. Le traitement des cibles est aussi simulé. Ainsi, on accumule les données brutes de cibles détectées aux têtes de grappe. Lorsque trois données détectent la même cible au même temps de détection, cette cible-temps devient prête à traiter. Le traitement est virtuel. À chaque seconde, la liste de cibles à traiter est réduite d'un certain nombre correspondant à la limite virtuelle de traitement du nœud. Cette valeur est toutefois arbitraire et ne traduit pas bien les capacités véritables d'un nœud occupé également à d'autres tâches telles que l'envoi et la réception de paquets et la gestion de la grappe.

Un autre facteur limitatif de notre solution est le débit de transmission des nœuds de détection. En raison du haut niveau de paquets transmis, un débit de 11 Mbps introduisait trop de paquets reçus avec des erreurs. Un débit de 1 Mbps a finalement été choisi. Il est à noter que tous les envois se faisaient avec un délai. Cette option de Qualnet vise à choisir un délai aléatoire situé dans une fenêtre de temps afin d'éviter que les paquets de tous les nœuds ne soient envoyés au même moment et pour réduire le nombre de collisions. Ce débit limité réduit le nombre de cibles pouvant être gérées par le système de détection.

5.3 Indications de recherches futures

Les possibilités de développements futurs sont assez variées. La principale modification qui pourrait être apportée consiste en intégrer une réelle simulation des cibles ainsi que de leur détection. Ceci permettrait de tester la solution proposée dans des situations plus fidèles à la réalité par leur variabilité. On pourrait par exemple faire varier la densité de cibles graduellement à travers le territoire disponible, en créant ainsi des engorgements locaux. On pourrait ainsi étudier la validité du protocole OSPF-wireless et son efficacité à déterminer les zones moins achalandées pour y acheminer une plus grande partie du trafic et de la charge de traitement. Il pourrait être intéressant de changer non seulement la distribution des cibles mais aussi la

distribution des nœuds de détection. En faisant notamment augmenter le nombre de nœuds de détection et le territoire disponible, on pourrait évaluer si pour un réseau de grande envergure la solution proposée serait plus avantageuse qu'AODV pour la transmission au serveur, et non plus seulement pour la transmission aux détecteurs. De plus, il serait utile de poursuivre l'investigation sur les nœuds limitrophes afin de resserrer les critères menant à cette qualification. Si le nombre de nœuds limitrophes partageant leurs données entre deux grappes était plus restreint, peut-être serait-il possible de localiser plus de cibles en évitant l'engorgement actuel de paquets ? La localisation de cibles par un réseau de détection sans-fil est un problème complexe qui requière l'utilisation de plusieurs concepts de réseautique pour assurer son succès. Le mémoire a présenté une approche proactive qui semble être adéquate pour un acheminement des cibles-temps aux nœuds détecteurs et qui offre des possibilités futures pour un acheminement efficace au serveur.

Références

AHRENHOLZ, J., HENDERSON, T., SPAGNOLO, P., BACCELLI, E., CLAUSEN, T. et JACQUET, P. (2004). Ospfv2 wireless interface type. Rapport technique, Boeing, Washington. Draft.

AKYILDIZ, I. F., WANG, X. et WANG, W. (2005). Wireless mesh networks : a survey. *Computer Networks*, 47, 445–487.

AL-KARAKI, J. N. et KAMAL, A. E. (2004). Routing techniques in wireless sensor networks : A survey. *IEEE Wireless Communications*, 11, 6–28.

AMIS, A. D. et PRAKASH, R. (2000). Load-balancing clusters in wireless ad hoc networks». Computer.ORG, éditeur, *3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology*. Texas, 25–32.

BECCHETTI, L., LEONARDI, S. et MUTHUKRISHNAN, S. (2004). Average stretch without migration. *Journal of Computer and System Sciences*, 68, 80–95.

BEUTEL, J. (2005). Robust topology formation using btnodes. *Computer Communications*, 28, 1523–1530.

BRENNER, P. (1996). A technical tutorial on the ieee 802.11 protocol. Rapport technique, Spread Spectrum Scene.

DE COUTO, D. S. J., AGUAYO, D., BICKET, J. et MORRIS, R. (2003). A high-throughput path metric for multi-hop wireless routing. ACM, éditeur, *International Conference on Mobile Computing and Networking,Proceedings of the 9th annual international conference on Mobile computing and networking*. San Diego, vol. 34, 134–146.

DRAVES, R., PADHYE, J. et ZILL, B. (2004). Comparison of routing metrics for static multi-hop wireless networks. *ACM SIGCOMM Computer Communication Review*, 34, 133–144.

GREEN, D. et OBAIDAT, M. (2003). Modeling and simulation of ieee802.11 wlan mobile ad hoc networks using topology broadcast reverse-path forwarding(tbrpf). *Computers Communications*, 26, 1741–1746.

HOPES, C. (2000). RFC 2992 : Analysis of an Equal-Cost Multi-Path Algorithm . Rapport technique, Next Hop Technologies, Mountain View.

IEEE-SA STANDARDS BOARD (2003). Part 11 : Wireless lan medium access control (mac) and physical layer (phy) specifications. Rapport technique, Standards IEEE ANSI/IEEE Std. 802.11, New-Jersey.

INETDAEMON (2004). Distance vector vs link state routing. Rapport technique, Inetdaemon.

LI, D.-C., WU, C. et CHANG, M. (2005). Determination of the parameters in the dynamic weighted round-robin method for network load balancing. *Computers & Operations Research*, 32, 2129–2145.

MOY, J. (1998). RFC 2328 :OSPF version 2 . Rapport technique, Ascend Communications Inc, Alameda.

NG, P. C. et LIEW, S. C. (2004). Offered load control in ieee802.11 multi-hop ad-hoc networks. *First IEEE International Conference on Mobile Ad-Hoc Sensor Systems Intelligent Robots and Systems (MASS 2004)*. Fort Lauderdale, 80–89.

NIEBERG, T. (2003). Distributed algorithms in wireless sensor networks. *Electronic Notes in Discrete Mathematics*, 13, 81–83.

OGIER, R., TEMPLIN, F. et LEWIS, M. (2004). RFC 3684 : Topology Dissemination Based on Reverse Path Forwarding . Rapport technique, SRI International, California.

PERKINS, C., BELDING-ROYER, E. et DAS, S. (2003). RFC 3561 : Ad hoc On-Demand Distance Vector (AODV) Routing . Rapport technique, Nokia Research Center, California.

ROYER, E. et TOH, C.-K. (1999). A review of current routing protocols for ad hoc mobile wireless networks. *Computer Networks*, 6, 46–55.

XU, Y. et QI, H. (2004). Distributed computing paradigms for collaborative processing in sensor networks. *Journal of Parallel and Distributed Computing*, 64, 945–959.

ZHOU, A. et HASSANEIN, H. (2001). Load-balanced wireless ad hoc routing. IEEExplore, éditeur, *Canadian Conference on Computer and Electrical Engineering*. Kingston, vol. 2, 1157–1161.